Stable XOR-based Policies for the Broadcast Erasure Channel with Feedback

Sophia Athanasiadou, Marios Gatzianas, *Member, IEEE*, Leonidas Georgiadis *Senior Member, IEEE* and Leandros Tassiulas, *Fellow, IEEE*

Abstract

In this paper we describe a network coding scheme for the Broadcast Erasure Channel with multiple stochastic flows, in the case of a single source transmitting packets to N users, where per-slot feedback is fed back to the transmitter in the form of ACK/NACK messages. This scheme uses only binary (XOR) operations and involves a network of virtual queues, along with special rules for coding packets and moving them among the queues that ensure instantaneous decodability. Based on the stabilizing scheduling policy described in the framework of [1], we show that, for $N = 4$ and i.i.d. erasure events, the network stability region of such a system effectively coincides with its information-theoretic capacity region, and provide a stabilizing policy that employs this XOR-based scheme.

I. Introduction

The information-theoretic capacity region of the Broadcast Erasure Channel (BEC) in the case of one transmitter and N multiple unicast sessions has been recently determined in [2] and [3]. Both of these papers proposed capacity achieving algorithms based on transmission of linear combinations of packets. However, these schemes are characterized by high complexity (as operations take place in a sufficiently large sized finite field) and decoding delay, since a sufficient number of linear combinations has to be received until a packet is decoded. In [4], we proposed a network coding scheme that overcomes these obstacles by using only XOR operations, generalizing the 2-user network coding scheme in [5] to the case of 3 users. Thus, two low complexity algorithms were proposed, namely `XOR1` and `XOR2`, that additionally had the advantageous property of "instantaneous decodability". By this term, it is meant that a receiver is able to decode packet p destined for it as soon as it receives an XOR combination of packets containing p. Algorithm `XOR2` was proved to achieve capacity for the case of i.i.d. channels as well as spatially independent channels with erasure probabilities that do not exceed 8/9.

However, the system considered in [4] is a saturated system, where a predefined number of packets needs to be transmitted to each user. This model is not frequently encountered in practice. Moreover, algorithms `XOR1` and `XOR2` cannot be easily generalized to more than 3 users. This happens because, at each time slot, coding choices have to be rigorously determined so that each transmission is optimally exploited in terms of allowing multiple users to simultaneously decode their packets as well as create favorable future coding opportunities. However, for $N > 3$, the number of coding choices increases dramatically so that there is no clear intuition on the optimal choice.

In the current work, we propose a general network coding scheme for the case of a single transmitter sending packets to N users through the BEC with feedback, generalizing the scheme proposed in [4]. Any packet arriving to the transmitter is placed in one of N queues. Depending on the received feedback, these packets (or XOR combinations of them) may travel through a network of virtual queues, before they reach their destination, in order to exploit the overhearing benefit of the broadcast channel. Coding and packet movement rules are imposed in order to ensure instantaneous decodability of packets and better exploitation of coding opportunities.

This research has been co-financed by the European Union (European Social Fund – ESF) and Greek national funds through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) – Research Funding Program: Thales. Investing in knowledge society through the European Social Fund. M. Gatzianas is supported by the ERC Starting Project Grant NOWIRE ERC-2009-StG-240317.

S. Athanasiadou and L. Georgiadis are with the Department of Electrical and Computer Engineering, Division of Telecommunications, Aristotle University of Thessaloniki, Thessaloniki, 54 124, Greece. M. Gatzianas is with the School of Computer and Communication Sciences, EPFL, Switzerland. L. Tassiulas is with the Computer Engineering and Telecommunications Department, University of Thessaly, Volos 38 221, Greece.

While in [4] we examined a saturated system, in this paper we consider the more realistic model where packets may arrive randomly at the transmitter at any time slot. Additionally, we use a backpressure type online algorithm that makes each coding choice based on instantaneous quantities, such as queue sizes, without requiring knowledge of future events. Therefore, we do not need to predefine the coding choices, and the proposed network coding scheme can be applied for an arbitrary number of users. For the specific case of 4 users and i.i.d. erasure events, we present a stabilizing policy on top of this network coding scheme and prove that the policy stability region coincides with the information theoretic capacity region of the standard BEC with feedback. This result is quite intriguing, considering that the proposed policy only uses a subset of all possible coding choices. Moreover, we address inplementation issues and present an overhead analysis for the maximum number of packet addresses that need to be included in the header of an XOR-coded packet.

The network stability of single hop broadcast erasure channesl with feedback has also been examined in [6], which considered broadcast traffic only and investigated the stability regions of plain retransmission and linear network coding schemes (parameterized over the field size) as opposed to a proposed dynamic virtual queue-based policy. The latter policy was shown to be optimal for 2 users while, for $N > 2$ and i.i.d. erasures, it achieved a stable rate that differs from the cut-set bound by a factor of $O(\epsilon^{m+1})$, where m is the number of queue "levels" that participate in the coding decision (see [6] for more details and definitions; m can be loosely regarded as a measure of the encoding complexity) and ϵ is the erasure probability. Although the structure of the virtual queues and coding rules are inspired by similar concepts as in our work, the actual rules for moving packets between the queues are much more involved in our work since we are interested in achieving the optimal stability region for all values of ϵ instead of only asymptotic optimality as $\epsilon \to 0$ (these notions of optimality ignore any overhead). An additional cause for rule complexity in our work is the fact that multiple unicast sessions are much more difficult to handle (due to the inherent competition between different sessions) than a single broadcast session. Furthermore, there is no guarantee in [6], for the general case of N users, regarding instantaneous decodability.

The work in [7] studied a network which is described by an underlying complete graph where each edge is modeled as a Markov chain ON/OFF channel (i.e. a generalization of the memoryless erasure channel), while there also exists a special "relay" node with XOR coding capabilities which can overhear all transmissions. Any transmissions to/from the relay are error-free. The work considers multiple unicast flows, originating in all nodes except for the relay, and explicitly accounts for instantaneous decodability by mapping this constraint into a specially constructed conflict graph (a similar graph structure is used in [**?**] to model the same constraint). It proposes an online backpressure policy that requires computing in each slot the maximum weight independent set of the time-varying conflict graph. Although the work bears similarities to our paper in terms of mathematical techniques and the optimization problem that results, the model is quite different. Hence, the proposed coding policies are quite different and the results in [7] cannot be used to show one of our main results, namely that the proposed scheduling and coding policies achieve channel capacity for BEC with i.i.d. erasures. In particular, the broadcast channel at the relay (which is the only node that can perform XOR coding) is error-free in [7], while we are interested in broadcast erasure channels.

In summary, the contribution of this paper lies in the development of a systematic framework for constructing instantaneously decodable feedback-based XOR coding schemes for the BEC with multiple unicast traffic and an arbitrary number of users. Although this framework provides general guidelines and determines the required properties of such a scheme, it does not specify all details of the coding procedure, thus allowing the code designer to trade complexity for performance. Additionally, combining this framework with the backpressure stabilizing policy in the model of [1], and carefully limiting the allowable coding choices, leads to the unexpected result that proper XOR combining is sufficient to achieve the capacity of the 4-user BEC with i.i.d. erasures.

The rest of the paper is organized as follows: in Section II, the system model is introduced along with some useful notation. In Section III, the proposed network coding scheme is described, while in Section IV the applied stabilizing policy is presented. In Section V, an outer bound on the stability region of the system under study is derived. In Section VI, we prove, for the case of 4 users and i.i.d. erasure events, that the stability region of such a system coincides with the capacity outer bound of the standard broadcast erasure channel with feedback. In Section VII we examine some implementation issues while Section VIII concludes the paper.

II. System model and notation

We describe some notation that will be used in the following. Sets are denoted by calligraphic letters, e.g. $\mathcal{M}$, and the empty set by $\emptyset$. The cardinality of set $\mathcal{M}$ is denoted by $|\mathcal{M}|$ and we write $M = |\mathcal{M}|$. Random variables are denoted by capital letters and their values by small case letters. Vectors are denoted by bold letters, e.g. $\boldsymbol{A} = (A_1, \ldots, A_n)$. The expected value of a random vector is the vector consisting of the expected values of its components, i.e., $\mathbb{E}[\boldsymbol{A}] = (\mathbb{E}(A_1), \ldots, \mathbb{E}(A_n))$.

We consider a time-slotted system where slot $t = 0, 1, \ldots$ corresponds to the time interval $[t,\, t+1)$. The system consists of a base station B and a set $\mathcal{N} = \{1, 2, \ldots, N\}$ of receivers (users). At the beginning of slot t, $A_i(t)$ data packets arrive at B with $\mathbb{E}(A_i) = \lambda_i$; these packets must be delivered to receiver i and are referred to as "flow i" packets, where we denote $\boldsymbol{A}(t) = (A_1(t), \ldots, A_N(t))$, $t = 0, 1, \ldots$. All packets consist of L bits with an individual transmission-time duration of 1 slot. A packet transmitted by B may be either correctly received or completely erased by any receiver (broadcast medium). After each transmission, the receivers send feedback to B (through an error-free zero-delay channel) informing whether the transmitted packet has been correctly received or not (ACK/NACK feedback). We also assume that if no packet is transmitted in a slot (idle slot), then all receivers realize that the slot is idle.

Packet arrivals are assumed to be independent and identically distributed across time, but arbitrarily correlated at a given time. That is, the process $\{\boldsymbol{A}(t)\}_{t=0}^{\infty}$ consists of i.i.d. random vectors, while the components of each vector $\boldsymbol{A}(t)$ may be arbitrarily correlated. Similarly, packet erasures are i.i.d across time and are initially assumed to be arbitrarily correlated at a given time (we later concentrate on the special case of spatially i.i.d. erasures). The packet arrival and erasure processes are independent. For subsets $\mathcal{S}, \mathcal{G} \subseteq \mathcal{N}$ with $\mathcal{S} \cap \mathcal{G} = \varnothing$, we denote by $P_{\mathcal{S},\mathcal{G}}$ the probability that a transmitted packet is erased at *all* receivers in $\mathcal{S}$ and received by *all* receivers in $\mathcal{G}$ (no condition is imposed on packet reception or erasure for receivers in $\mathcal{N} - (\mathcal{S} \cap \mathcal{G})$). We also denote by $\epsilon_{\mathcal{S}}$ the probability that a transmitted packet is erased by all receivers in $\mathcal{S}$, i.e., $\epsilon_{\mathcal{S}} = P_{\mathcal{S},\varnothing}$. For simplicity, we slightly abuse the notation and write ϵ_i or ϵ_{ij} instead of $\epsilon_{\{i\}}$ or $\epsilon_{\{i,j\}}$, respectively.

III. Network coding scheme description

Exogenous packets arriving at B and being intended for user $i \in \mathcal{N}$ are called "native packets for i". A packet is simply termed "native" if it is a native packet for some user (due to the unicast traffic, a packet is native for exactly one user). According to the policies to be described below, all transmitted packets are either native, or XOR combinations of native packets. In other words, any transmitted packet p can be written as $p = \bigoplus_{l=1}^{n} q_l$, where q_l are native packets and we say that "p contains q_l". It is possible, and actually beneficial, for p to contain native packets for more than one user. To shorten the description in the following, we say that a packet p is an XOR combination of native packets even when p consists of a single native packet. The following definitions will be crucial in the subsequent analysis.

Definition 1. User i is a *listener* of a packet p iff both of the following conditions are true:

1) p is an XOR combination of packets, not necessarily native, that i has correctly received.
2) p contains no native packet for i that is unknown to i. Equivalently, if p contains a native packet q for user i, then the packet q is known to (i.e. has already been decoded by) i.

Notice that the second condition of Definition 1 does not assert that p always contains a native packet q for user i, only that the existence of such a packet implies that q is known to i. To illustrate this definition, let a packet p be of the form $p = p_1 \oplus p_2$ where p_1, p_2 are native packets (not necessarily intended for user $i \in \mathcal{N}$). Then, i is a listener for p if either p_1 and p_2 have both been received by i, or if p itself has been received by i and any of p_1 and p_2 that *may* be native packets for i are already known to i.

Definition 2. User i is a *destination* of a packet p if either p is a native packet intended for user i that has not been received yet by (i.e. is unknown to) i, or if p can be decomposed as an XOR combination of the form $p = q \oplus c$ where

1) q is a native packet intended for i and unknown to i, and
2) i is a *listener* of c.

We hereafter use the term *destination* to exclusively refer to the above technical definition. Notice that the decomposition of a packet p with *destination* i alluded to in Definition 2 is unique, since c cannot contain an unknown native packet for i due to the second condition of Definition 1 (recall that i is also a *listener* of c). We also denote the unique q in p as $q = p(i)$ and call $p(i)$ the "unknown packet" of i in p. As will be seen, transmitted packets may have several receivers as *destinations* or *listeners*. As a final note, it is apparent that the notions of *destination* and *istener* are time-varying, as each user receives new packets over time.

The main features of the policies to be presented are as follows.

Basic Algorithmic Features

1) Any transmitted packet is an XOR combination of native packets.
2) If a transmitted packet (equivalently, an XOR combination) p contains a native packet q intended for $i \in \mathcal{N}$, which is unknown to i, then i is a *destination* for p. Hence, since according to the definition of *destination* it holds either $p = q$ or $p = q \oplus c$, where c is known to i, user i can immediately decode q as $q = p \oplus c$.

Under the proposed policies, packets may be placed in various virtual queues, based on the received feedback. A general virtual queue $Q_{\mathcal{D}}^{\mathcal{L}}$ is characterized by two index sets $\mathcal{L}, \mathcal{D}$ satisfying the following condition:

Condition I

1) $\mathcal{L}, \mathcal{D} \subseteq \mathcal{N}$,
2) $\mathcal{L} \cap \mathcal{D} = \emptyset$,
3) $\mathcal{D} \neq \emptyset$,
4) $\mathcal{L} = \emptyset$ only if $|\mathcal{D}| = 1$.

For simplicity, we will denote queue $Q_{\{i,j\}}^{\{k\}}$ by Q_{ij}^{k}, and queue $Q_{\{i\}}^{\emptyset}$ by Q_i. Also, we use the notation $p_{\mathcal{D}}^{\mathcal{L}}$ to denote a packet that is stored in virtual queue $Q_{\mathcal{D}}^{\mathcal{L}}$. The policies ensure that the following properties holds for any packet $p \in Q_{\mathcal{D}}^{\mathcal{L}}$:

Basic Properties of packets $p \in Q_{\mathcal{D}}^{\mathcal{L}}$:

1) Any receiver $i \in \mathcal{D}$ is a *destination* for p and any $i \in \mathcal{L}$ is a *listener* of p.
2) If p contains a native packet for user $i \notin \mathcal{D}$, then this native packet has already been decoded by (i.e. is known to) i. Equivalently, p contains no unknown native packet for any user in $\mathcal{D}^c$, where ^{c} denotes set complement.
3) For any native packet q for user i that has not yet been decoded by i, there exists exactly one packet $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ (for some $\mathcal{L}$, $\mathcal{D}$) such that $q = p(i)$.

Hence, each packet $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ can be written as $p = \oplus_{i \in \mathcal{D}} p(i) \oplus d$. We note that these queues are virtual in the sense that there are N real queues Q_i^R at B, where queue Q_i^R, $i \in \mathcal{N}$, contains native packets intended for user i that are unknown to this user. When a packet of flow $i \in \mathcal{N}$ arrives at B, it is placed in queue Q_i^R. The virtual queues can be tracked by maintaining appropriate pointers to the real queues. Hence, at any instance in time, we can partition the real queue Q_i^R into disjoint queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$, $i \in \mathcal{D}$, where packet $q \in Q_i^R$ is also in $Q_{\mathcal{D}}^{\mathcal{L}}(i)$ if for some $p \in Q_{\mathcal{D}}^{\mathcal{L}}$ it holds $q = p(i)$. Therefore, viewing each queue as a set of packets, we have

$$Q_i^R = \bigcup_{\mathcal{L}, \mathcal{D}} Q_{\mathcal{D}}^{\mathcal{L}}(i)$$

where the union is over all sets $\mathcal{L}, \mathcal{D}$ satifying Condition I.

The virtual queues describe in effect the information that is gathered (through feedback) during the system operation for each of the packets in these queues, and are helpful in the description of the policies to be presented. The real queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$ will be useful in the analysis performed in Sections V, VI. We classify virtual queues into N levels, where level $w \in \{1, \ldots, N\}$ contains all queues $Q_{\mathcal{D}}^{\mathcal{L}}$ such that $|\mathcal{L} \cup \mathcal{D}| = w$. Moreover, we classify queues of level $w \geq 3$ into sublevels, where sublevel $w.u$ includes queues of level w with $|\mathcal{L}| = u$, $u \in \{1, \ldots, w-1\}$. In Figure 1, we give an example of virtual and real queues when $N = 3$.

In general, during the operation of the system, XOR combinations of packets are transmitted, at most one from each of the queues $Q_{\mathcal{D}}^{\mathcal{L}}$. While the specific choice of packets depends on the received feedback and the specific algorithm that is employed, the following rule always holds.

Basic Coding Rule

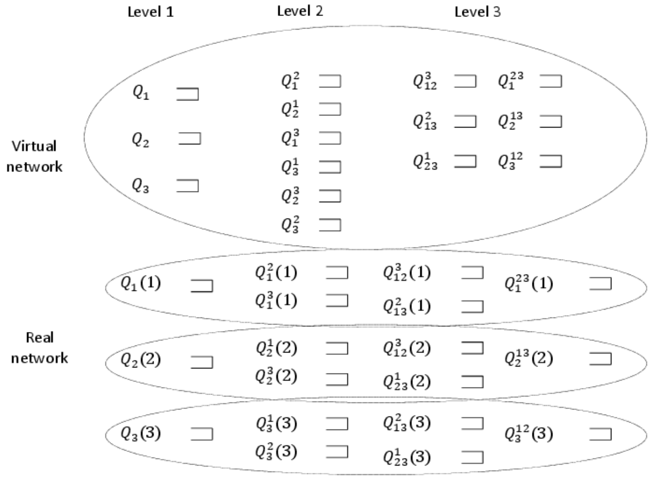

Figure 1. Virtual and real queues.

A set $\mathcal{P} = \left\{ p_{\mathcal{D}_1}^{\mathcal{L}_1}, \dots, p_{\mathcal{D}_m}^{\mathcal{L}_m} \right\}$ of $m = |\mathcal{P}|$ packets, *one* from each of the *different* virtual queues $\left\{ Q_{\mathcal{D}_1}^{\mathcal{L}_1}, \dots, Q_{\mathcal{D}_m}^{\mathcal{L}_m} \right\}$, can be combined (by XORing) into a single coded packet only if

$$\mathcal{D}_n \subseteq \mathcal{L}_r, \ \forall r \neq n, \ n, r \in \{1, \dots, m\}. \tag{1}$$

In other words, the destination set of each packet is a subset of the others packets' listener sets.

Note that the Basic Coding Rule implies that $\mathcal{D}_n \cap \mathcal{D}_r = \emptyset$, for all $r \neq n$, n, $r \in \{1, \dots, m\}$. Indeed, $i \in \mathcal{D}_n$ implies, through (1), that $i \in \mathcal{L}_r$ and, since according to Condition I it holds $\mathcal{D}_r \cap \mathcal{L}_r = \emptyset$, it follows that $i \notin \mathcal{D}_r$. In fact, Condition I and the Basic Coding Rule imply that any user $i \in \cup_{n=1}^{m} \mathcal{D}_n$ that receives the transmitted XOR combination will be able to decode its native packet.

Lemma 3. *Assume that Algorithmic Features 1, 2 are satisfied at the beginning of slot t. Then, if the packet to be transmitted at slot t is composed using the Basic Coding Rule, Algorithmic Features 1, 2 are also satisfied at slot t+1.*

Proof: The first part of the Algorithmic Features (i.e. any transmitted packet is a XOR combination of native packets) is trivial so we concentrate on the second part and assume that the transmitted packet $p = \bigoplus_{k=1}^{m} p_{\mathcal{D}_k}^{\mathcal{L}_k}$ contains some unknown native packet q for user $i \in \mathcal{N}$ (i.e. q is contained in some of the packets $p_{\mathcal{D}_k}^{\mathcal{L}_k}$). By the Basic Properties, it must hold $i \in \cup_{k=1}^{m} \mathcal{D}_k$, otherwise q would be known to i. Since $\mathcal{D}_k$ are also disjoint sets, it follows that there exists exactly one $k^* \in \{1, \dots, m\}$ such that $i \in \mathcal{D}_{k^*}$ and q is contained in $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$. But now $i \in \mathcal{D}_{k^*}$ is a destination for $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}} = q \oplus c$, where q is the unknown native packet for i and i is *listener* of c. The Basic Coding Rule implies that $i \in \mathcal{L}_r$, for all $r \neq k^*$, so that it holds $p = q \oplus c \oplus \bigoplus_{r \neq k^*} p_{\mathcal{D}_r}^{\mathcal{L}_r}$. Finally, $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$ cannot contain an unknown native packet for i other than q, since i is a destination for $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$. Hence, i is a destination for p and the proof is complete. ■

Next, we describe how packets are moved between virtual queues $Q_{\mathcal{D}}^{\mathcal{L}}$ (and, consequently, between the corresponding real queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$), based on the received feedback. Generally, packets may be placed in various queues such that the Basic Properties are satisfied. For example, assume that packet $p = p_{12}^3 \oplus p_3^{12}$ is transmitted (this combination satisfies the Basic Coding Rule), and that only user 2 receives the packet; hence packet $p_{12}^3(2)$ reaches its destination and is removed from the real queue $Q_{12}^3(2)$. For the other packet movements, two choices are consistent with Basic Properties:

1) Packet p_{12}^3 is moved to virtual queue Q_1^{23} and packet p_3^{12} is not moved - hence, regarding the real queues, only packet $p_{12}^3(1)$ is moved to queue $Q_1^{23}(1)$. This is consistent with Basic Properties since, after receiving p, receiver 2 becomes a listener for $p_{12}^3 = p \oplus p_3^{12}$, while receiver 3 is already a listener for that packet.

2) Packet p is moved to queue $Q_{13}^{2}$and packet p_{12}^{3}, p_{3}^{12} are removed from queues Q_{12}^{3}, Q_{3}^{12} respectively - hence, packet $p_{12}^{3}(1)$ is moved to queue $Q_{13}^{2}(1)$ and $p_{3}^{12}(3)$ is moved to queue $Q_{13}^{2}(3)$.

Intuitively, the higher the level (and, within the same level, the higher the sublevel) of a virtual queue in which a packet p is stored, the better are the chances of combining the packet with other packets in such a way that multiple successful receptions are effected with a single transmission. Hence, when multiple choices for packet movement arise, we select the one that ensures that all packets involved in a transmission are placed in a higher level (and within the same level higher sublevels), else they are not moved. Hence, in the example above, we choose the first option. This is the underlying rationale on which packet movement is based in the following description.

Rules for Packet Movement: Let packet p of the form $p = p_{\mathcal{D}_1}^{\mathcal{L}_1} \oplus \ldots \oplus p_{\mathcal{D}_m}^{\mathcal{L}_m}$ satisfying the Basic Coding Rule be chosen for transmission, and let $\mathcal{S}$ be the set of users that receive p (the packet is erased at the rest of the receivers). If $\mathcal{S} = \emptyset$, the packet is retransmitted. Otherwise, $\mathcal{S} \neq \emptyset$ and let $\tilde{\mathcal{S}}$ be the subset of receivers in $\mathcal{S} \cap (\cup_{k=1}^{m} \mathcal{L}_k)$ with the following property: $i \in \tilde{\mathcal{S}}$ iff it belongs to at least $m-1$ of the sets $\mathcal{L}_k,\ k \in \{1, \ldots, m\}$. Hence, before transmission of p, user $i \in \tilde{\mathcal{S}}$ is a listener for all but at most one of the packets $p_{\mathcal{D}_k}^{\mathcal{L}_k}\ k \in \{1, \ldots, m\}$.

Every receiver in $\mathcal{S}$ which is a destination of p can decode its corresponding packet i.e., for any $k \in \{1, \ldots, m\}$ and $i \in \mathcal{D}_k \cap \mathcal{S}$, packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ is delivered to destination i. This is performed as follows: the Basic Coding Rule implies that $i \in \mathcal{L}_r$ for all $r \neq k$. Hence, this $i \in \mathcal{D}_k \cap \mathcal{S}$ can compute $p_{\mathcal{D}_k}^{\mathcal{L}_k} = p \oplus \bigoplus_{r \neq k} p_{\mathcal{D}_r}^{\mathcal{L}_r}$, where all RHS terms are known to i after transmission of p (p is received correctly by i, since $i \in \mathcal{S}$). But $i \in \mathcal{D}_k \cap \mathcal{S}$ is also a destination for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$, so that it can decode the unique unknown native packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ contained in $p_{\mathcal{D}_k}^{\mathcal{L}_k}$. Notice that any $i \in \mathcal{D}_k \cap \mathcal{S}$ becomes a listener for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ after receiving p.

If $\mathcal{D}_k - \mathcal{S} = \emptyset$, i.e. all users in $\mathcal{D}_k$ receive p, then packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ is removed from virtual queue $Q_{\mathcal{D}_k}^{\mathcal{L}_k}$. Hence, if $\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S} = \emptyset$ all packets are delivered to their destination and are removed from the corresponding queues. Otherwise, $\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S} \neq \emptyset$ and we distinguish the following cases:

1) it holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k) = \emptyset$, equivalently, $\mathcal{S} \subseteq \cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k)$. Notice that, for $m > 1$, the latter condition is equivalent, by the Basic Coding Rule, to $\mathcal{S} \subseteq \cup_{k=1}^{m} \mathcal{L}_k$, while for $m = 1$ it reduces to $\mathcal{S} \subseteq \mathcal{L}_1 \cup \mathcal{D}_1$. In both cases, and for each $k \in \{1, \ldots, m\}$, packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}$, $\text{where}\, \mathcal{D}_k - \mathcal{S} \neq \emptyset$, is moved to queue $Q_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$ and each packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i),\ i \in \mathcal{D}_k - \mathcal{S}$, to queue $Q_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}(i)$. The consistency of these packet movements with Basic Properties is subsequently proved in Lemma 4. Hence, according to this rule, a packet is either not moved (if $\mathcal{D}_k \cap \mathcal{S} = \emptyset$), or is moved to a higher level (or within the same level but higher sublevel) queue.
2) it holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k) \neq \emptyset$. Again, this condition is equivalent to $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m} \mathcal{L}_k \neq \emptyset$, for $m > 1$, and $\hat{\mathcal{S}} = \mathcal{S} - (\mathcal{L}_1 \cup \mathcal{D}_1) \neq \emptyset$ for $m = 1$. We further distinguish two subcases:
    a) If $|(\cap_{k=1}^{m} \mathcal{L}_k \cup \mathcal{S}) \cup (\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S})| > \max_{k=1,\ldots,m} |\mathcal{L}_k \cup \mathcal{D}_k|$,[1]then packet p is moved to $Q_{\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^{m} \mathcal{L}_k \cup \mathcal{S}}$, packets $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ are removed from queues $Q_{\mathcal{D}_k}^{\mathcal{L}_k}$ and for each i in $\mathcal{D}_k - \mathcal{S}$ packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ is moved from queue $Q_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ to queue $Q_{\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^{m} \mathcal{L}_k \cup \mathcal{S}}(i)$. Lemma 4 shows again that this packet movement is consistent with Basic Properties and the packets are moved only to higher level or sublevel queues.
    b) If $|(\cap_{k=1}^{m} \mathcal{L}_k \cup \mathcal{S}) \cup (\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S})| \leq \max_{k=1,\ldots,m} |\mathcal{L}_k \cup \mathcal{D}_k|$ then
        i) if $\mathcal{S} \cap (\cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k)) = \emptyset$, no further action is taken.
        ii) else, set $\mathcal{S} \leftarrow \mathcal{S} \cap (\cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k))$ and follow the rules of case 1.

The validity of the above actions is proved in the following result.

Lemma 4. *Assuming that the Basic Properties are satisfied at the beginning of slot t, then the application of the Basic Coding Rule and Rules for Packet Movement to the packet transmitted at slot t satisfies the Basic Properties at the beginning of slot $t+1$.*

Proof: Let the transmitted packet p at slot t have the form $p = \bigoplus_{k=1}^{m} p_{\mathcal{D}_k}^{\mathcal{L}_k}$ where the various $\mathcal{L}_k$, $\mathcal{D}_k$ satisfy the Basic Coding Rule. We use the notation of the Rules for Packet Movement and examine each case of the Rules separately. The case $\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S} = \emptyset$ (equivalently, $\cup_{k=1}^{m} \mathcal{D}_k \subseteq \mathcal{S}$) is trivial, since no packets are moved between queues (hence, the Basic Properties still hold at $t+1$) and all users in $\cup_{k=1}^{m} \mathcal{D}_k$ decode their unknown native packets, as explained in the Rules for Packet Movement. We now assume $\cup_{k=1}^{m} \mathcal{D}_k - \mathcal{S} \neq \emptyset$ and, for brevity,

[1] it is easy to verify that this inequality is alwaystrue for $m = 1$.

only examine in detail the case $m > 1$, since $m = 1$ can be handled as a special case. Hence, we assume $m > 1$ and further distinguish cases as follows:

1) It holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k) = \emptyset$, i.e. $\mathcal{S} \subseteq \cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k) = \cup_{k=1}^{m}\mathcal{L}_k$. In this case, for any $k = 1, \ldots, m$ such that $\mathcal{D}_k - \mathcal{S} \neq \emptyset$, the packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ is moved to virtual queue $Q_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$ so that we need to show the following properties (shown in bold face) to be true at the beginning of slot $t+1$:
   - **All users in the non-empty set $\mathcal{D}_k - \mathcal{S}$ are *destinations* for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ and all users in $\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}$ are *listeners* for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$**: the *destination* property follows immediately by the Basic Properties at slot t, since $\mathcal{D}_k - \mathcal{S} \subseteq \mathcal{D}_k$. For the *listener* property, we examine each set in the union $\mathcal{L}_k \cup (\mathcal{D}_k \cap S) \cup \tilde{\mathcal{S}}$ separately: again, by the Basic Properties at t, all users in $\mathcal{L}_k$ are *listeners* of $p_{\mathcal{D}_k}^{\mathcal{L}_k}$. Similarly, any user $i \in \mathcal{D}_k \cap \mathcal{S}$ knows the value of $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ as well as the native packet intended for i (see the explanation in the Rules for Packet Movement). Notice that $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ cannot contain a second unknown native packet for i, since $i \in \mathcal{D}_k \cap \mathcal{S}$ is also a *destination* for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$. Hence, any $i \in \mathcal{D}_k \cap \mathcal{S}$ is a *listener* of $p_{\mathcal{D}_k}^{\mathcal{L}_k}$. Finally, consider any user in $\tilde{\mathcal{S}} \cap \mathcal{L}_k^c \cap (\mathcal{D}_k \cap \mathcal{S})^c = \tilde{\mathcal{S}} \cap \mathcal{L}_k^c \cap \mathcal{D}_k^c$ (since it holds $\tilde{\mathcal{S}} \subseteq \mathcal{S}$) where we concentrate on users which do not belong to $\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S})$. By construction of $\tilde{\mathcal{S}}$, any user $i \in \tilde{\mathcal{S}} \cap \mathcal{L}_k^c \cap \mathcal{D}_k^c$ belongs to all sets $\mathcal{L}_r$, for $r \neq k$, since it does not belong to $\mathcal{L}_k$ and it must belong to at least $m-1$ of the $\mathcal{L}$ sets. Hence, since this user received p, it can deduce $p_{\mathcal{D}_k}^{\mathcal{L}_k} = p \oplus \bigoplus_{r \neq k} p_{\mathcal{D}_r}^{\mathcal{L}_r}$. Furthermore, since $i \in \mathcal{D}_k^c$, the Basic Properties at t imply that $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ cannot contain an unknown native packet for i, so that i is again a *listener* of $p_{\mathcal{D}_k}^{\mathcal{L}_k}$.
   - **Packet $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ contains no unknown native packets for any user in set $(\mathcal{D}_k - \mathcal{S})^c$**: consider any user $i \in (\mathcal{D}_k - \mathcal{S})^c = \mathcal{D}_k^c \cup \mathcal{S} = \mathcal{D}_k^c \cup (\mathcal{S} \cap \mathcal{D}_k)$. The desired property now follows immediately from the Basic Properties at t (i.e. $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ contains no unknown native packets for any user in $\mathcal{D}_k^c$) and the fact that all users in $\mathcal{D}_k \cap \mathcal{S}$ become *listeners* for $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ after successfull reception of p.
   - **For each native packet q that has not been decoded by its intended user, there exists exactly one packet $\hat{p} \in Q_{\mathcal{D}}^{\mathcal{L}}$, for some $\mathcal{L}$, $\mathcal{D}$, such that $q = \hat{p}(i)$**: to prove this property, we use the fact that it is true at the beginning of slot t and, after transmission at slot t occurs, the only native packets that are decoded at the end of slot t are those intended for users in $\cup_{k=1}^{m}\mathcal{D}_k \cap \mathcal{S}$. Hence, it suffices to only consider any unknown native packets contained in $p_{\mathcal{D}_k}^{\mathcal{L}_k}$, for all $k \in \{1, \ldots, m\}$ with $\mathcal{D}_k - \mathcal{S} \neq \emptyset$ at the beginning of slot t. These native packets remain unknown to their destinations at the beginning of slot $t+1$; however, due to the movement of $p_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ to $Q_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}(i)$, these native packets are still included in exactly one packet, namely $p_{\mathcal{D}_k - \mathcal{S}}^{\mathcal{L}_k \cup (\mathcal{D}_k \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, and the proof is complete.

2) It holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m}(\mathcal{L}_k \cup \mathcal{D}_k) \neq \emptyset$, equivalently $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^{m}\mathcal{L}_k$. We only examine case 2.(a) of the Rules of Packet Movement, since case 2.(b) reverts to case 1, which has already been dealt with. Specifically, we assume $|(\cap_{k=1}^{m}\mathcal{L}_k \cup \mathcal{S}) \cup (\cup_{k=1}^{m}\mathcal{D}_k - \mathcal{S})| > \max_{k=1,\ldots,m}|\mathcal{L}_k \cup \mathcal{D}_k|$ so that the composite packet p is moved to $Q_{\cup_{k=1}^{m}\mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^{m}\mathcal{L}_k \cup \mathcal{S}}$. Note that, applying verbatim the argument used in Case 1, we can show that all users in $\cup_{k=1}^{m}\mathcal{D}_k \cap S$ decode their unknown native packets. We now need to prove the following for the beginning of slot $t+1$.
   - **All users in $\cup_{k=1}^{m}\mathcal{D}_k - \mathcal{S}$ are *destinations* for p and all users in $\cap_{k=1}^{m}\mathcal{L}_k \cup \mathcal{S}$ are *listeners* for p.** For the *destination* property, consider any $i \in \cup_{k=1}^{m}\mathcal{D}_k - \mathcal{S}$. Then, there exists some $k^* \in \{1, \ldots, m\}$ such that $i \in \mathcal{D}_{k^*} - \mathcal{S}$ and, by the Basic Coding Rule, $i \in \mathcal{L}_r$ for all $r \neq k^*$. Since $i \in \mathcal{D}_{k^*}$ is a *destination* for $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$ (Basic Properties at t) and we can write $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}} = q \oplus c$, where q is an unknown native packet for i and i is *listener* of c, we conclude that $p = q \oplus c \oplus \bigoplus_{r \neq k^*} p_{\mathcal{D}_r}^{\mathcal{L}_r}$ where all RHS terms (except q) are known to i. Since neither $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$, nor any of $p_{\mathcal{D}_r}^{\mathcal{L}_r}$ can contain any other unknown packet for i (recall that i is a *listener* for all $p_{\mathcal{D}_r}^{\mathcal{L}_r}$ with $r \neq k^*$), it follows that i is a *destination* of p. To prove the *listener* property, consider any $i \in \cap_{k=1}^{m}\mathcal{L}_k \cup \mathcal{S}$. If $i \in \cap_{k=1}^{m}\mathcal{L}_k$, then i already knows the value of p since it knows each individual component $p_{\mathcal{D}_k}^{\mathcal{L}_k}$, for $k = 1, \ldots, m$. Furthermore, no $p_{\mathcal{D}_k}^{\mathcal{L}_k}$ can contain an unknown native packet for i since i is a *listener* for all of these packets (due to Basic Properties at t); hence, i is a *listener* of p. It remains to consider the case $i \in \mathcal{S} - \cap_{k=1}^{m}\mathcal{L}_k$ (notice that any such i learns the value of p upon reception). For any $i \in \mathcal{S} - \cap_{k=1}^{m}\mathcal{L}_k$, i.e. $i \in \mathcal{S} \cap (\cup_{k=1}^{m}\mathcal{L}_k^c)$, there exists some k^* such that $i \in \mathcal{S} \cap \mathcal{L}_{k^*}^c$. But the Basic Coding Rule implies that $\mathcal{L}_{k^*} \supseteq \mathcal{D}_r \Rightarrow \mathcal{L}_{k^*}^c \subseteq \mathcal{D}_r^c$ for all $r \neq k^*$ so that

$i \in \mathcal{S} \cap \mathcal{L}_{k^*}^c \cap \mathcal{D}_r^c$. By the Basic Properties at t, no packet $p_{\mathcal{D}_r}^{\mathcal{L}_r}$ with $r \neq k^*$ can contain an unknown native packet for i, since $i \in \mathcal{D}_r^c$. Hence, any unknown native packet for i can only be contained in $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$ provided that $i \in \mathcal{D}_{k^*}$ (if $i \notin \mathcal{D}_{k^*}$, the Basic Properties at t again guarantee that $p_{\mathcal{D}_{k^*}}^{\mathcal{L}_{k^*}}$ cannot contain an unknown native packet for i). But then, the conditions $i \in \mathcal{D}_{k^*}$and $i \in \mathcal{S}$ imply that i has already decoded its native packet, so that we can eliminate this case and conclude that any $i \in \mathcal{S} - \cap_{k=1}^m \mathcal{L}_k$ is a *listener* of p.

- **Packet p contains no unknown native packets for any users outside set $\cup_{k=1}^m \mathcal{D}_k - \mathcal{S}$.** Consider any $i \in (\cup_{k=1}^m \mathcal{D}_k - \mathcal{S})^c = \mathcal{S} \cup \cap_{k=1}^m \mathcal{D}_k^c$. The property now follows immediately from the Basic Properties at t and the fact that all users in $\mathcal{S}$ become *listeners* of p after receiving it.
- **For each native packet q that has not been decoded by its intended user, there exists exactly one packet $\hat{p} \in Q_{\mathcal{D}}^{\mathcal{L}}$, for some $\mathcal{L}$, $\mathcal{D}$, such that** $q = \hat{p}(i)$: the proof is similar to Case 1 by considering the users in $\cup_{k=1}^m \mathcal{D}_k \cap \mathcal{S}$ and examining the native packet movement from $Q_{\mathcal{D}_k}^{\mathcal{L}_k}(i)$ to $Q_{\cup_{k=1}^m \mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^m \mathcal{L}_k \cup \mathcal{S}}(i)$.

■

Example 5. Suppose packet $p = p_1^{2346} \oplus p_{24}^{135} \oplus p_3^{1246}$ is transmitted, so $m = 3$ and $\mathcal{D}_1 = \{1\}, \mathcal{D}_2 = \{2,4\}, \mathcal{D}_3 = \{3\}$, $\mathcal{L}_1 = \{2,3,4,6\}$, $\mathcal{L}_2 = \{1,3,5\}$, $\mathcal{L}_3 = \{1,2,4,6\}$.

- Suppose p is received by users $2, 5$ and 6, so $\mathcal{S} = \{2,5,6\}$. It holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k) = \{2,5,6\} - \{1,2,3,4,5,6\} = \emptyset$, so we are in case 1. We have $\mathcal{S} \cap \left(\cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k)\right) = \mathcal{S} = \{2,5,6\}$ and $\tilde{\mathcal{S}} = \{2,6\}$ because user 5 does not belong to $m-1=2$ sets $\mathcal{L}_k$ but only to set $\mathcal{L}_2$. The 3 packets are moved as follows:
  - packet p_1^{2346} is not moved because $\mathcal{D}_1 \cap \mathcal{S} = \{1\} \cap \{2,5,6\} = \emptyset$ (equivalently, it is moved to $Q_{\mathcal{D}_1 - \mathcal{S}}^{\mathcal{L}_1 \cup (\mathcal{D}_1 \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, i.e. $Q_{\{1\}}^{\{2,3,4,6\} \cup \emptyset \cup \{2,6\}} = Q_1^{2346}$).
  - packet p_{24}^{135} is moved to $Q_{\mathcal{D}_2 - \mathcal{S}}^{\mathcal{L}_2 \cup (\mathcal{D}_2 \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, i.e. $Q_{\{2,4\} - \{2,5,6\}}^{\{1,3,5\} \cup (\{2,4\} \cap \{2,5,6\}) \cup \{2,6\}} = Q_4^{12356}$.
  - packet $p_3^{1246}$is not moved because $\mathcal{D}_3 \cap \mathcal{S} = \{3\} \cap \{2,5,6\} = \emptyset$ (equivalently, it is moved to $Q_{\mathcal{D}_3 - \mathcal{S}}^{\mathcal{L}_3 \cup (\mathcal{D}_3 \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, i.e. $Q_{\{3\}}^{\{1,2,4,6\} \cup \emptyset \cup \{2,6\}} = Q_3^{1246}$).
- Suppose now that p is received by users 7 and 8, so $\mathcal{S} = \{7,8\}$. It holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k) = \{7,8\} - \{1,2,3,4,5,6\} = \{7,8\} \neq \emptyset$, so we are in case 2. We have
$$\left|\left(\cap_{k=1}^3 \mathcal{L}_k \cup \mathcal{S}\right) \cup \left(\cup_{k=1}^3 \mathcal{D}_k - \mathcal{S}\right)\right| =$$
$$|((\{2,3,4,6\} \cap \{1,3,5\} \cap \{1,2,4,6\}) \cup \{7,8\}) \cup ((\{1\} \cup \{2,4\} \cup \{3\}) - \{7,8\})| = |\{1,2,3,4,7,8\}| = 6.$$
We also have
$$\max_{k=1,\ldots,3} |\mathcal{L}_k \cup \mathcal{D}_k| = \max\{|\{1,2,3,4,6\}|, |\{1,2,3,4,5\}|, |\{1,2,3,4,6\}|\} = 5.$$
Therefore, we are in subcase (a), and p is moved to $Q_{\cup_{k=1}^3 \mathcal{D}_k - \mathcal{S}}^{\cap_{k=1}^3 \mathcal{L}_k \cup \mathcal{S}}$, i.e. Q_{1234}^{78}.
- If p is received by user 7, then $\mathcal{S} = \{7\}$. It holds $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k) = \{7\} - \{1,2,3,4,5,6\} = \{7\} \neq \emptyset$, so we are in case 2. We have $\left|\left(\cap_{k=1}^3 \mathcal{L}_k \cup \mathcal{S}\right) \cup \left(\cup_{k=1}^3 \mathcal{D}_k - \mathcal{S}\right)\right| = |\{1,2,3,4,7\}| = 5$ and $\max_{k=1,\ldots,3} |\mathcal{L}_k \cup \mathcal{D}_k| = 5$. We also have $\mathcal{S} \cap \left(\cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k)\right) = \{7\} \cap \{1,2,3,4,5,6\} = \emptyset$, therefore we are in subcase (b.i) and no packets are moved.
- If p is received by users 2 and 7, then $\mathcal{S} = \{2,7\}$. We have $\hat{\mathcal{S}} = \mathcal{S} - \cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k) = \{2,7\} - \{1,2,3,4,5,6\} = \{7\} \neq \emptyset$, so we are in case 2. We have $\left|\left(\cap_{k=1}^3 \mathcal{L}_k \cup \mathcal{S}\right) \cup \left(\cup_{k=1}^3 \mathcal{D}_k - \mathcal{S}\right)\right| = |\{1,2,3,4,7\}| = 5$ and $\max_{k=1,\ldots,3} |\mathcal{L}_k \cup \mathcal{D}_k| = 5$. We also have $\mathcal{S} \cap \left(\cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k)\right) = \{2,7\} \cap \{1,2,3,4,5,6\} = \{2\} \neq \emptyset$, therefore we are in subcase (b.ii). Next, we set $\mathcal{S} \leftarrow \mathcal{S} \cap \left(\cup_{k=1}^3 (\mathcal{L}_k \cup \mathcal{D}_k)\right)$, i.e. $\mathcal{S} \leftarrow \{2\}$, and follow the rules of case 1. We have $\tilde{\mathcal{S}} = \{2\}$ and the 3 packets are moved as follows:
  - packet p_1^{2346} is not moved because $\mathcal{D}_1 \cap \mathcal{S} = \{1\} \cap \{2\} = \emptyset$ (equivalently, it is moved to $Q_{\mathcal{D}_1 - \mathcal{S}}^{\mathcal{L}_1 \cup (\mathcal{D}_1 \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, i.e. $Q_{\{1\}}^{\{2,3,4,6\} \cup \emptyset \cup \{2\}} = Q_1^{2346}$).
  - packet p_{24}^{135} is moved to $Q_{\mathcal{D}_2 - \mathcal{S}}^{\mathcal{L}_2 \cup (\mathcal{D}_2 \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, i.e. $Q_{\{2,4\} - \{2\}}^{\{1,3,5\} \cup (\{2,4\} \cap \{2\}) \cup \{2\}} = Q_4^{1235}$.
  - packet $p_3^{1246}$is not moved because $\mathcal{D}_3 \cap \mathcal{S} = \{3\} \cap \{2\} = \emptyset$ (equivalently, it is moved to $Q_{\mathcal{D}_3 - \mathcal{S}}^{\mathcal{L}_3 \cup (\mathcal{D}_3 \cap \mathcal{S}) \cup \tilde{\mathcal{S}}}$, i.e. $Q_{\{3\}}^{\{1,2,4,6\} \cup \emptyset \cup \{2\}} = Q_3^{1246}$).

As mentioned above, the rules for packet movement specified above are not unique. Also, information (who knows what about each packet) may be lost with this movement (see case 2.(b)).

As explained, this choice is made on intuitive grounds in order to keep the system manageable and amenable to analysis. However, as will be seen in the next Section, for $N = 4$ even a more restrictive choice of rules suffices to implement a policy with asymptotically (as packet length increases) maximal stability region when the channel erasure probabilities are i.i.d.

We next present two Lemmas, which follow from the Basic Coding Rule and Rules for Packet Movement, that will be useful in Section VII. Lemma 6 states that a packet p, which is an XOR combination of packets that are stored in queues of at most level k, can include at most k such packets.

Lemma 6. *Consider packet $p = p_{\mathcal{D}_1}^{\mathcal{L}_1} \oplus \ldots \oplus p_{\mathcal{D}_m}^{\mathcal{L}_m}$, where $|\mathcal{D}_i \cup \mathcal{L}_i| \le k$, $i \in \{1, \ldots, m\}$. Then $m \le k$.*

Proof: It holds $|\mathcal{D}_1 \cup \mathcal{L}_1| \le k$. Also, since $\mathcal{D}_r \subseteq \mathcal{L}_1,\ \forall r \in \{2, \ldots, m\}$, we have $\bigcup_{r=2}^{m} \mathcal{D}_r \subseteq \mathcal{L}_1$ and $\bigcup_{r=1}^{m} \mathcal{D}_r \subseteq \mathcal{D}_1 \cup \mathcal{L}_1$. Since $\mathcal{D}_i \cap \mathcal{D}_j = \emptyset,\ i \neq j$, it holds

$$\sum_{r=1}^{m} |\mathcal{D}_r| = \left| \bigcup_{r=1}^{m} \mathcal{D}_r \right| \le |\mathcal{D}_1 \cup \mathcal{L}_1| \le k \tag{2}$$

Since $\mathcal{D}_r \neq \emptyset,\ r \in \{1, \ldots, m\}$, i.e, $|\mathcal{D}_r| \ge 1$,we conclude that $\sum_{r=1}^{m} |\mathcal{D}_r| \ge m$, which together with (2) implies $m \le k$, as desired. ■

Let packet $p = p_{\mathcal{D}_1}^{\mathcal{L}_1} \oplus \ldots \oplus p_{\mathcal{D}_m}^{\mathcal{L}_m}$ be transmitted. Then, depending on the feedback from the users and in accordance to the Rules for Packet Movement, either p is moved as a whole to a virtual queue, or packets $p_{\mathcal{D}_1}^{\mathcal{L}_1}, \ldots, p_{\mathcal{D}_m}^{\mathcal{L}_m}$ are moved separately to virtual queues. More precisely, the following Lemma follows immediately from the Rules for Packet Movement.

Lemma 7. *After transmission of a packet at slot t, let packet p (not necessarily the transmitted packet) be placed at a virtual queue of level k.n. Then, either a) p is a combination of packets that at the beginning of slot t were at queues of level less that k, or b) p is a copy of a packet that at the beginning of slot t was either at level r, $r \in \{0, \ldots, k-1\}$, or at sublevel $k.l$, $1 \le l \le n-1$.*

IV. Stabilizing Scheduling Policy

In this Section, we investigate the design of policies that, under the coding restrictions and packet movements described in Section III, stabilize the system whenever possible. We first need some definitions.

A. System Stability and Stability Region

Let $X(t)$, $t = 0, 1, \ldots$ be a stochastic process.

Definition 8. [Stability] The process $X(t)$, $t = 0, 1, \ldots$ is stable iff

$$\lim_{q \to \infty} \limsup_{t \to \infty} \Pr\left(X(t) > q\right) = 0.$$

Consider next a time-slotted system $\mathcal{U}$. At the beginning of each slot, a number of new packets belonging to a set $\mathcal{N}$ of "flows" arrive to the system. Newly arriving packets of flow $i \in \mathcal{N}$ are placed at infinite size queues, i.e. no incoming packets are ever dropped. These packets are processed by a policy π belonging to a set Π of admissible policies. During this processing, packets are moved between a set of queues $\mathcal{Q}$, and, at some time, they exit the system. Under policy π, let $Q_l^{\pi}(t)$ be the number of packets in queue $Q_l \in \mathcal{Q}$ at time t, and define $\hat{Q}^{\pi}(t) = \sum_{Q_l \in \mathcal{Q}} Q_l^{\pi}(t)$. Let also $A_i(t)$ $i \in \mathcal{N}$ be the number of flow i packets arriving at the beginning of slot t. For the purposes of this paper, we assume that the process $\{\boldsymbol{A}(t)\}_{t=0}^{\infty}$, where $\boldsymbol{A}(t) = \{A(t)\}_{i \in \mathcal{N}}$, consists of i.i.d vectors with $\boldsymbol{\lambda} = \mathbb{E}[\boldsymbol{A}(t)]$.

Definition 9. [System Stability]

1) For a given arrival rate vector λ, system $\mathcal{U}$ is stable under policy π if the process $\hat{Q}^{\pi}(t)$ is stable.

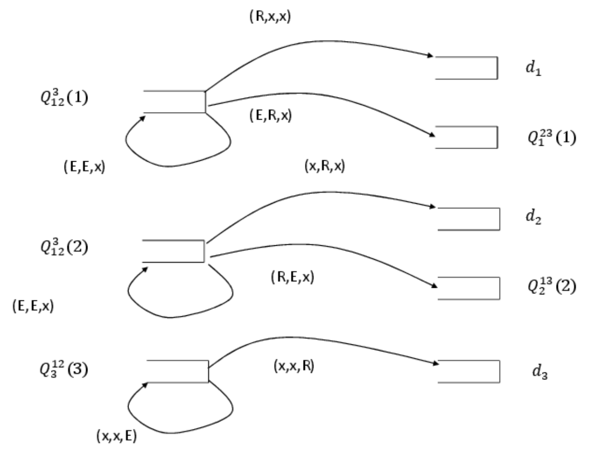

Figure 2. Possible movements of packets $p^3_{12}(1)$, $p^3_{12}(2)$, $p^{12}_3(3)$. Destination of user i is denoted as d_i. Received feedback is denoted as (u_1, u_2, u_3), where u_i is the feedback from user i, where R, E stand for received, erased, respectively, while X denotes an indifferent value (either R or E).

2) The stability region of a policy, $\mathcal{R}^\pi$, $\pi \in \Pi$, is the closure of the set of arrival rates for which $\mathcal{U}$ is stable under π.
3) The stability region $\mathcal{R}_\Pi$ of system $\mathcal{U}$ under the set of policies Π is the closure of the set $\cup_{\pi\in\Pi}\mathcal{R}^\pi$.
4) A policy $\pi^* \in \Pi$ is stabilizing if $\mathcal{R}_\Pi = \mathcal{R}^{\pi^*}$.

Consider now the system under study in the current work. At the beginning of each slot, a decision must be made at the base station concerning the combination of packets from the virtual queues that must be XORed to form the packet $p = p^{\mathcal{L}_1}_{\mathcal{D}_1} \oplus \ldots \oplus p^{\mathcal{L}_m}_{\mathcal{D}_m}$ to be transmitted. Such a decision is called a "control" $I^{\mathcal{L}_1,\ldots,\mathcal{L}_m}_{\mathcal{D}_1,\ldots,\mathcal{D}_m}$ and we denote the set of such controls by $\mathcal{I}$. Notice that, by definition, a control is identified by the set of pairs $\{(\mathcal{D}_i, \mathcal{L}_i)\}^m_{i=1}$ and not by their order, i.e., control $I^{\mathcal{L}_1,\ldots,\mathcal{L}_m}_{\mathcal{D}_1,\ldots,\mathcal{D}_m}$ is identical to control $I^{\mathcal{L}_{\sigma(1)},\ldots,\mathcal{L}_{\sigma(m)}}_{\mathcal{D}_{\sigma(1)},\ldots,\mathcal{D}_{\sigma(m)}}$ for any permutation $\sigma(i)$ of the indicies.

We assume henceforth that the Basic Coding Rule is followed for the formation of packet p. For this system, an admissible policy consists of selecting, at the beginning of each time slot, one of the available controls $I^{\mathcal{L}_1,\ldots,\mathcal{L}_m}_{\mathcal{D}_1,\ldots,\mathcal{D}_m}$ to form a packet p for transmission. After p is transmitted, packets are moved among the real queues $Q^{\mathcal{L}_k}_{\mathcal{D}_k}(i)$ according to the Rules for Packet Movement described in Section III.

A characteristic of such movements is that the destination of a packet movement is random and determined by the feedback received after packet transmission. For example, assume that $N = 3$ and control $I^{3,12}_{12,3}$ is taken, i.e. packet $p = p^3_{12} \oplus p^{12}_3$ is transmitted. The packets involved in this transmission are $p^3_{12}(1)$, $p^3_{12}(2)$, $p^{12}_3(3)$. Figure 2 shows the possible movements of these packets according to the received feedback.

Our objective in this Section is to design a stabilizing policy for the system described above. It turns out that this system falls in the class of systems whose stability has been studied in [1]. We next summarize the formulation and main results in [1] that will be useful in the development that follows.

Consider a slotted-time network with a node set $\mathcal{M} \cup d$ and edge (i.e. link) set $\mathcal{E}$, where the special node d represents the destination of traffic originated at the nodes in $\mathcal{M}$. The node set $\mathcal{M}$ is identified with the set of all real queues $Q^{\mathcal{L}}_{\mathcal{D}}(k)$, with $k \in \mathcal{D}$, i.e. each node $m \in \mathcal{M}$ corresponds to exactly one real queue. Let $\mathcal{E}^m_{out}$, $\mathcal{E}^m_{in}$ denote, respectively, the set of outgoing links and incoming links to node m. We allow self-loops in the network, i.e. for node $m \in \mathcal{M}$, there may be a link (m, m), implying that the sets $\mathcal{E}^m_{out}$, $\mathcal{E}^m_{in}$ both contain node m. A finite set of controls $\mathcal{I}$ is available. For each control $I \in \mathcal{I}$, "transmission" takes place over the sets of outgoing links of node $m \in \mathcal{N}$, $\mathcal{E}^m_{out}$, as follows.

- If, at a given slot, control $I \in \mathcal{I}$ is applied, then for any node $m \in \mathcal{M}$ at most $\hat{\mu}_m(I) \geq 0$ packets may be transmitted over the set $\mathcal{E}^m_{out}$ in the following random manner:
  - For each $m \in \mathcal{M}$ and $I \in \mathcal{I}$, there is a random sequence $\{R^m_n(I)\}^\infty_{n=1}$, where each $R^m_n(I)$ takes values in the set $\mathcal{E}^m_{out}$, with the following interpetation. The n-th packet transmitted over the set $\mathcal{E}^m_{out}$ when control

I is applied, is received *only* by the recipient of the link $R_n^m(I)$. Of course, if $R_n^m(I) = (m,m)$ then the packet is not received by any node in $\mathcal{E}_{out}^m$-$\{m\}$, hence it remains at node m.

For a given n and I, the random variables $\left\{R_n^i(I)\right\}_{i\in\mathcal{N}}$ may be arbitrarily correlated. Moreover, we assume that for each control $I \in \mathcal{I}$, the random sequences $\left\{\{R_n^m(I)\}_{m\in\mathcal{M}}\right\}_{n=1}^{\infty}$ are i.i.d and we denote $p_e^m(I) \triangleq \Pr(R_n^m(I) = e)$, $e \in \mathcal{E}_{out}^m(I)$, so that

$$\sum_{e\in\mathcal{E}_{out}^m} p_e^m(I) = 1 \tag{3}$$

To describe the stability region $\mathcal{R}_\Pi$ of this network, we need some preliminary definitions. For control $I \in \mathcal{I}$, we define the set $\Gamma(I)$ of vectors $\boldsymbol{f}$,

$$\Gamma(I) = \left\{\boldsymbol{f} = \{f_e\}_{e\in\mathcal{E}} : \ f_e = p_e^m(I)\mu_m,\ 0 \le \mu_m \le \hat{\mu}_m(I),\ i \in \mathcal{N}, e \in \mathcal{E}_{out}^m\right\}, \tag{4}$$

and the convex hull $\mathcal{H}$ of the sets $\Gamma(I)$ as

$$\mathcal{H} = \operatorname{conv}\left(\Gamma(I), I \in \mathcal{I}\right) \tag{5}$$

The stability region of the network is described by the following Theorem.

Theorem 10. *[1] The stability region $\mathcal{R}_\Pi$ of the system is the set of arrival rates $\boldsymbol{\lambda} = \{\lambda_m\}_{m\in\mathcal{M}}$, $\lambda_m \ge 0$, for which there exists a vector $\mathbf{f} \in \mathcal{H}$ such that for any node $m \in \mathcal{M}$ it holds,*

$$\sum_{e\in\mathcal{E}_{in}^m} f_e + \lambda_m \le \sum_{e\in\mathcal{E}_{out}^m} f_e. \tag{6}$$

We will apply the formulation described above to the network consisting of the real queues $Q_{\mathcal{D}}^{\mathcal{L}}(i),\ i \in \mathcal{D}$, i.e., we consider $\mathcal{M} = \left\{Q_{\mathcal{D}}^{\mathcal{L}}(i),\ i \in \mathcal{D}\right\}$. For this network, since only one packet is transmitted per slot from any queue m, we have $\hat{\mu}_m(I) = 1,\ m \in \mathcal{M}$. Also, the packet transition probabilities $p_e^m(I)$ can be easily calculated. An example is given below.

Example 11. Consider the case $N = 3$ and assume that control $I_{12,3}^{3,12}$ is chosen, hence a combination $p = p_{12}^3 \oplus p_3^{12}$ is transmitted, where $p_{12}^3 = p_{12}^3(1) \oplus p_{12}^3(2)$ and $p_3^{12} = p_3^{12}(3)$. The transition probabilities are then as follows:

- Packet $p_{12}^3(1)$.
  1) If p is received by user 1, $p_{12}^3(1)$ is removed from $Q_{12}^3(1)$ and delivered to d_1 (to d for the equivalent network). This event has probability $P_{\emptyset,\{1\}}$.
  2) If p is erased at user 1 and received by user 2, packet p_{12}^3 is moved to virtual queue and $p_{12}^3(1)$ is moved to queue $Q_1^{23}(1)$. This event has probability $P_{\{1\},\{2\}}$.
  3) If p is erased at users 1 and 2, $p_{12}^3(1)$remains at $Q_{12}^3(1)$. This event has probability $P_{\{1,2\},\emptyset}$.
- Packet $p_{12}^3(2)$. The transition probabilities are determined as in the previous case, by interchanging the indices 1,2.
- Packet $p_3^{12}(3)$.
  1) If p is received by user 3, $p_3^{12}(3)$ is removed from $Q_3^{12}(3)$ and delivered to d_3. This event has probability $P_{\emptyset,\{3\}}$.
  2) If p is erased at 3, $p_3^{12}(3)$ remains at $Q_3^{12}(3)$. This event has probability $P_{\{3\},\emptyset}$.

The only difference between the network $(\mathcal{M} \cup \{d\}, \mathcal{E})$ and our model is that, in the latter, there are N packet destinations, $d_i,\ i \in \mathcal{N}$ (one for each of the receivers) instead of a single one. However, we can combine all these destinations to a single destination d, so that any packet arriving in d_i is considered to arrive at d. This affects neither the admissible policies, nor the queue sizes at the various native queues at the base station. Hence, system stability is not affected, provided that we are interested in the total queue size at the base station.

Two implementation issues are worth mentioning at this point. First, there must be a mechanism for the receivers to know the constituents of the XOR combination of each received packet, in order to be able to use this packet in the decoding process. The simplest way to implement this is to use packet addresses to identify the packets involved in the XOR combination of the transmitted packet. These addresses can be placed in the packet header. Reserving bits to describe packet addresses implies some loss of throughput due to the introduced overhead. To simplify the description, in the current and next Section we do not take the overhead into account and address the issue of stability in packets per slot. In Section VII, we discuss the number of addressed needed and loss of throughput due to overhead.

The second issue is that under the schemes described in Section III, the receivers need to save received packets so that they can correctly decode at a later time. Hence, if we are interested in taking these queues into consideration, we must ensure that the system remains stable even if the sizes of these queues are added to the total queue size at the base station. In fact, if the receivers are never informed by the base station as to which of their received packets will not be needed in the future, it is easy to devise scenarios where the queue sizes at the receivers grow to infinity even though the queues at the base station are stable (notice that system stability is only defined with respect to transmitter side queues). A simple way to deal with this problem is described in Section VII.

B. *Stabilizing Policy*

Applying directly the results in [1], we obtain the stabilizing policy described below. At the beginning of each time slot, the policy chooses a control of the form $I = I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m} \in \mathcal{I}$, where all real queues in the set $\tilde{\mathcal{M}}\left(I\right) = \bigcup\limits_{l=1}^{m} \bigcup\limits_{k\in\mathcal{D}_l} \left\{ Q_{\mathcal{D}_l}^{\mathcal{L}_l}\left(k\right) \right\}$ are nonempty, and forms the appropriate packet to be transmitted on that slot, $p = \oplus_{l=1}^{m} p_{\mathcal{D}_l}^{\mathcal{L}_i}$. If control I is chosen, one packet from each of the real queues in the set $\tilde{\mathcal{M}}\left(I\right)$ may be moved to another queue inside the network, or may reach the destination (and thus exit the network). No packets from any of the other queues are moved. The algorithm for choosing the appropriate control is the following.

Algorithm 1 At each decision slot :

1) For each control $I = I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m} \in \mathcal{I}$:
   - Form the weights
   $$c_i\left(I\right) = \max\left\{ K_i - \sum_{e=\{i,j\}\in\mathcal{E}_{out}^{i}} p_e^i\left(I\right) K_j,\ 0 \right\},\ i \in \tilde{\mathcal{M}}\left(I\right),$$
   where K_i is the length of the queue corresponding to node i.
   - Form the reward under the given control,
   $$C\left(I\right) = \sum_{i\in\tilde{\mathcal{M}}(I)} c_i\left(I\right).$$
2) Select the control that maximizes the reward, i.e. $I^* = \arg\max{}_{I\in\mathcal{I}} C\left(I\right)$.

V. Outer Bound on the Stability Region

In this Section we derive an outer bound on the stability region of the system under study by deparameterizing (i.e. eliminating the flow variables $\boldsymbol{f}$ in) Theorem 10. This bound is identical with the bound on the information-theoretic capacity region of the broadcast erasure channel with feedback presented in [2], [3]. Although it was shown in [8] that the capacity region of the system under consideration is the same as the stability region of the system, we cannot directly invoke this result to derive the stability region outer bound via the capacity outer bound in [2], [3]. The reason is that the latter capacity bound does not take into account the case of slots without any packet transmission, i.e. idle slots, so that, in principle, coding algorithms may take advantage of idle slots to increase capacity beyond the outer bound in [2], [3]. To distinguish the two channels, we call the broadcast channel studied in [2], [3] the "standard" broadcast channel, and refer to the channel under study in this paper (i.e. the one containing idle slots) as the "extended" broadcast channel.

As will be seen, the capacity of the standard broadcast channel, measured in information bits per transmitted symbol, differs from the capacity of the extended broadcast channel by at most 1 bit; in fact, this difference decreases exponentially w.r.t. the packet length L. Specifically, the following Theorem is proved in the Appendix.

Theorem 12. *A capacity outer bound $\mathcal{C}_{out}$, measured in bits per transmitted symbol, for the N-user "extended" broadcast erasure channel with feedback is given by*

$$\mathcal{C}_{out} = \left\{ \boldsymbol{R} : \max_{\sigma\in\mathcal{P}} \left(\sum_{k\in\mathcal{N}} \frac{R_k}{1-\epsilon_{\{\sigma(1),\ldots,\sigma(k)\}}} - 2^{-L/A_\sigma} A_\sigma \right) \le L \right\} \tag{7}$$

where $\mathcal{P}$ is the set of all permutations on $\mathcal{N}$ and $A_\sigma = \sum_{k\in\mathcal{N}} \frac{1}{1-\epsilon_{\{\sigma(1),\ldots,\sigma(k)\}}}$.

We first describe the stability region of Theorem 10 in a form that is more convenient for calculations. Any $\boldsymbol{f}$ in $\mathcal{H}$ can be written in the form

$$\boldsymbol{f} = \sum_{I\in\mathcal{I}} \phi_I \boldsymbol{f}\left(I\right), \quad \text{for some } \phi_I \geq 0,\ \sum_{I\in\mathcal{I}} \phi_I \leq 1, \tag{8}$$

where

$$\boldsymbol{f}\left(I\right) = \left(f_e\left(I\right)\right)_{e\in\mathcal{E}},$$
$$f_e\left(I\right) = p_e^m(I)\mu_m(I),\ 0 \leq \mu_m\left(I\right) \leq \hat{\mu}_m(I),\ m\in\mathcal{M}, e\in\mathcal{E}_{out}^m,$$

and, for any control $I = I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m}$,

$$\hat{\mu}_m(I) = \left\{ \begin{array}{ll} 1 & \text{if } m\in\tilde{\mathcal{M}}\left(I\right) \\ 0 & \text{otherwise} \end{array} \right. . \tag{9}$$

Hence it holds,

$$\begin{aligned} \sum_{e\in\mathcal{E}_{out}^m} f_e &= \sum_{e\in\mathcal{E}_{out}^m} \sum_{I\in\mathcal{I}} \phi_I f_e\left(I\right) \\ &= \sum_{I\in\mathcal{I}} \phi_I \sum_{e\in\mathcal{E}_{out}^m} p_e^m(I)\mu_m\left(I\right) \end{aligned} \tag{10}$$

and

$$\begin{aligned} \sum_{e\in\mathcal{E}_{in}^m} f_e &= \sum_{e\in\mathcal{E}_{in}^m} \sum_{I\in\mathcal{I}} \phi_I f_e\left(I\right) \\ &= \sum_{I\in\mathcal{I}} \sum_{e=(l,m)\in\mathcal{E}_{in}^m} \phi_I \mu_l\left(I\right) p_e^l(I). \end{aligned} \tag{11}$$

Replacing (10), (11) in (6), we have

$$\sum_{I\in\mathcal{I}} \phi_I \left(\sum_{e=(l,m)\in\mathcal{E}_{in}^m} \mu_l\left(I\right) p_e^l(I) \right) + \bar{\lambda}_m \leq \sum_{I\in\mathcal{I}} \phi_I \left(\sum_{e\in\mathcal{E}_{out}^m} p_e^m(I)\mu_m\left(I\right) \right), \quad m\in\mathcal{M} \tag{12}$$

or equivalently, taking into account (3),

$$\sum_{I\in\mathcal{I}} \phi_I \left(\sum_{\substack{e=(l,m)\in\mathcal{E}_{in}^m \\ l\neq m}} \mu_l\left(I\right) p_e^l(I) \right) + \bar{\lambda}_m \leq \sum_{I\in\mathcal{I}} \left(1 - p_{(m,m)}^m(I)\right) \mu_m\left(I\right)\phi_I, \quad m\in\mathcal{M} \tag{13}$$

Hence, the stability region $\mathcal{R}_\Pi$ of the system is described by either one of (12), (13), combined with

$$0 \leq \mu_m\left(I\right) \leq \hat{\mu}_m(I) \tag{14}$$
$$\phi_I \geq 0 \tag{15}$$
$$\sum_{I\in\mathcal{I}} \phi_I \leq 1 \tag{16}$$

where $\hat{\mu}_m\left(I\right)$ is given by (9). Since, in our case, new packet arrivals are only placed in queues $Q_i(i)$ $i\in\mathcal{N}$, it holds

$$\bar{\lambda}_m = \left\{ \begin{array}{ll} \lambda_i & \text{if} \quad m = Q_i(i),\ i\in\mathcal{N} \\ 0 & \text{otherwise} \end{array} \right. \tag{17}$$

The next Theorem, which is proved in Appendix A, describes the main result of this Section.

Theorem 13. *The following relation holds*

$$\mathcal{R}_\Pi \subseteq \left\{ \boldsymbol{\lambda}:\ \max_{\sigma\in\mathcal{S}} \sum_{i=1}^{4} \frac{\lambda_{\sigma(i)}}{1-\epsilon_{\tilde{\sigma}(i)}} \leq 1 \right\} \triangleq \mathcal{C}_u, \tag{18}$$

where $\mathcal{P}$ is the set of permutations on $\mathcal{N}$ and $\tilde{\sigma}(i) = \{\sigma\left(1\right),\ldots,\sigma(i)\}$.

Table I
PERMITTED CONTROLS FOR LEVELS 1 TO 4.

	Level 1	Level 2	Level 3	Level 4
Permitted controls	Control	Control	Control	Control
	I_i	$I_{i,j}^{j,i}$	$I_{i,jk}^{jk,i}$	$I_{i,jkl}^{jkl,i}$
		I_i^j	I_{jk}^i	I_{jkl}^i
			$I_{i,j,k}^{jk,ik,ij}$	$I_{ij,kl}^{kl,ij}$
			$I_{i,j}^{jk,ik}$	$I_{ij,k,l}^{kl,ijl,ijk}$
			I_i^{jk}	I_{ij}^{kl}
				$I_{i,j,k,l}^{jkl,ikl,ijl,ijk}$
				$I_{i,j,k}^{jkl,ikl,ijl}$
				$I_{i,j}^{jkl,ikl}$
				I_i^{jkl}

VI. THE CASE OF I.I.D. CHANNELS: STABILITY REGION FOR 4 USERS

In this Section, we assume that the erasure events for all receivers are i.i.d, and denote by ϵ the probability of such an event. Hence,

$$P_{\mathcal{S},\mathcal{G}} = \epsilon^{|\mathcal{S}|} \left(1-\epsilon\right)^{|\mathcal{G}|}.$$

We consider the case of a channel with 4 receivers and show that, for all $0 \leq \epsilon < 1$, if $\boldsymbol{\lambda} \in \mathcal{C}_u$, then $\boldsymbol{\lambda} \in \mathcal{R}_\Pi$, i.e. $\mathcal{R}_\Pi \supseteq \mathcal{C}_u$. Hence, in this case we have $\mathcal{R}_\Pi = \mathcal{C}_u$ and the stability region using only XOR operations coincides (barring addressing overhead) with the capacity region of the standard broadcast channel, and is within one bit from the stability region of the extended broadcast channel (which, due to [8], is equal to the capacity region of the extended broadcast channel in Theorem 12).

To proceed, we restrict the set of available controls by allowing only intra-level coding, i.e. we only combine packets from queues of the same level. This restriction simplifies the calcuations and shows that even a restricted set of controls suffices to achieve the maximal stability region when channel erasure events are i.i.d. We note however, that if channel statistics are non-i.i.d., the additional controls are helpful in increasing the stability region.

The set of permitted controls is described in Table I, where $i, j, k, l \in \{1, 2, 3, 4\}$ are distinct.

For the rest of this Section, we assume without loss of generality that

$$\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4, \tag{19}$$

which implies that

$$\mathcal{C}_u = \left\{ \boldsymbol{\lambda} : \sum_{i=1}^{4} \frac{\lambda_i}{1-\epsilon^i} \leq 1 \right\} \tag{20}$$

We will show that if $\boldsymbol{\lambda} \in \mathcal{C}_u$, then $\boldsymbol{\lambda} \in \mathcal{R}_\Pi$, which is equivalent to solving the following problem for any $0 \leq \epsilon < 1$.

Problem: If $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$ and $\sum_{i=1}^{4} \frac{\lambda_i}{1-\epsilon^i} \leq 1$, find parameters ϕ_I satisfying (13)-(16), where $\mathcal{M}$ is the set of all queues $Q_{\mathcal{D}}^{\mathcal{L}}(i)$, $i \in \mathcal{D}$, and $\mathcal{L}, \mathcal{D}$ satisfy Condition I.

In the following, we will describe the procedure according to which $\mu_m(I)$, ϕ_I, $m \in \mathcal{M}$, $I \in \mathcal{I}$, are calculated. First, we set

$$\mu_m(I) = \hat{\mu}_m(I), \; m \in \mathcal{M}, \; I \in \mathcal{I}, \tag{21}$$

ensuring that (14) is satisfied. It remains to determine ϕ_I, $I \in \mathcal{I}$. Notice that, for any given value of ϵ, (21) transforms (13), (15), (16) into a linear program (LP) w.r.t ϕ_I, so that achievability of the rate $\boldsymbol{\lambda}$ is reduced to LP feasibility (a similar LP-based approach is used to describe an achievable scheme for a 2 user MIMO setting over broadcast erasure channels in [9]). However, since ϵ takes a continuum of values, we cannot solve the resulting LP for each ϵ but need to determine ϕ_I analytically.

To simplify the notation somewhat, for control $I = I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m}$ we denote

$$\phi_I = \phi_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m}.$$

An overview of the approach follows. We start from inequalities (13) referring to queues at level 1, i.e. $Q_i\left(i\right)$, and determine all ϕ_i, ensuring that these inequalities are satisfied. In general, having determined ϕ_I for all controls that involve queues at level l, we consider the inequalities (13) referring to queues at level $l+1$ and determine ϕ_I for all controls that involve queues at level $l+1$, ensuring that these inequalities are satisfied. During this process, it is ensured that (15) is satisfied. After all ϕ_I are computed, it is checked that (16) is also satisfied.

We now proceed with the detailed description of the manner in which $\phi_I,\ I \in \mathcal{I}$ are determined. We will use the following terminology in the decription. If, under an allowable control I, it is possible to have a packet movement from queue i to j, we say that there is a "flow from queue i to queue j" under control I and we name $p_{(i,j)}^{i}\left(I\right)$, the "probability of flow" from i to j under control I. We also say that there is "flow from queue i to queue j" if it is possible to have a packet movement from queue i to queue j under some of the allowable controls.

Level 1 At this level, there are 4 queues (equivalently, nodes in $\mathcal{M}$) of the form $Q_i\left(i\right),\ i \in \{1,\ldots,4\}$. There are no incoming flows from other nodes to $Q_i\left(i\right)$, but there are new packet arrivals of rate λ_i at every $Q_i\left(i\right)$. The only control that may result in packets leaving $Q_i\left(i\right)$ is I_i, so inequality (13) becomes $\lambda_i \leq (1-\epsilon^4)\cdot\phi_i$. To satisfy this inequality, we set

$$\phi_i = \lambda_i/(1-\epsilon^4),\ \forall i \in \{1,\ldots,4\}\,. \tag{22}$$

Level 2 At level 2, there are 12 queues of the form $Q_i^j\left(i\right),\ i,j \in \{1,\ldots,4\}\,,\ i \neq j$. The only incoming flow to each of these nodes is under control I_i, with probability $\epsilon^3(1-\epsilon)$, while there are two outgoing flows, under controls $I_{i,j}^{j,i}$ and I_i^j, that result in packets leaving with probability $1-\epsilon^3$. Hence, inequality (13) becomes

$$\epsilon^3(1-\epsilon)\cdot\phi_i \leq (1-\epsilon^3)\cdot\phi_{i,j}^{j,i} + (1-\epsilon^3)\cdot\phi_i^j. \tag{23}$$

Similarly, for node $Q_j^i\left(j\right)$ we have

$$\epsilon^3(1-\epsilon)\cdot\phi_j \leq (1-\epsilon^3)\cdot\phi_{i,j}^{j,i} + (1-\epsilon^3)\cdot\phi_j^i. \tag{24}$$

Since ϕ_i, ϕ_j have already been determined by (22), there are now 2 inequalities with 3 unknown parameters, $\phi_{i,j}^{j,i}$, ϕ_i^j, ϕ_j^i. Next, assuming $i \leq j$ we set

$$\phi_j^i = 0,\ \text{if}\ i \leq j. \tag{25}$$

Intuition for this is gained from inequality (19). Specifically, observe first that through (8), we can interpret ϕ_I as the long term proportion of time that control I is applied. Also, it can be seen that as long as both queues $Q_i^j\left(i\right),\ Q_j^i\left(j\right)$ are nonempty, the algorithm in Section IV-B chooses to apply control $I_{i,j}^{j,i}$ instead of I_i^j or I_j^i, since control $I_{i,j}^{j,i}$ potentially results in two packet transitions during one time slot, while I_i^j, I_j^i can lead to only one packet transition. During the time interval that control $I_{i,j}^{j,i}$ is applied, the same number of packets is expected to have left queues Q_i^j and Q_j^i, as erasure probabilities are identical. When one of Q_i^j, Q_j^i is empty, then either I_j^i or I_i^j, respectively, is applied. Since we assumed $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \lambda_4$ and $i \leq j$, it is more likely that Q_j^i empties while $I_{i,j}^{j,i}$ is applied, and this event becomes very likely as queue sizes increase. Therefore, in the long run, one expects that control I_j^i is never applied and this is expressed by (25). The intuition for setting certain ϕ_I to zero in the following description follows a similar reasoning. To satisfy (24), we set

$$\phi_{i,j}^{j,i} = \epsilon^3(1-\epsilon)\cdot\lambda_j/(1-\epsilon^3)\cdot(1-\epsilon^4), \quad i \leq j. \tag{26}$$

We substitute $\phi_{i,j}^{j,i}$ in (23) and, to satisfy (23), we set

$$\phi_i^j = \left(\lambda_i - \lambda_j\right)\epsilon^3\left(1-\epsilon\right)/\left(1-\epsilon^3\right)\left(1-\epsilon^4\right), \quad i \leq j. \tag{27}$$

Observe that (25), (27) can be combined to

$$\phi_i^j = \left[\lambda_i - \lambda_j\right]^+\epsilon^3\left(1-\epsilon\right)/\left(1-\epsilon^3\right)\left(1-\epsilon^4\right), \tag{28}$$

where $[x] \triangleq \max\{x, 0\}$. It is easy to check that the quantities in (26)-(28) are nonnegative.

Level 3 At this level, there are 12 virtual queues of type Q_{ij}^{k} (corresponding to real queues $Q_{ij}^{k}(i)$ and $Q_{ij}^{k}(j)$) and 12 queues of type Q_i^{jk} (corresponding to real queues $Q_i^{jk}(i)$), where $i, j, k \in \{1, \ldots, 4\}$,$i \neq j \neq k$.

- Incoming flow to Q_{ij}^{k} (respectively, to both $Q_{ij}^{k}(i)$ and $Q_{ij}^{k}(j)$) occurs under control $I_{i,j}^{j,i}$ with probability $\epsilon^3(1-\epsilon)$. Outgoing flows from nodes of this form occur under controls $I_{ij,k}^{k,ij}$ and I_{ij}^{k}, with probability $1-\epsilon^3$. While for each of the queues $Q_{ij}^{k}(i)$ and $Q_{ij}^{k}(j)$ there is one inequality of the form (13), it turns out that these inequalities are identical. Hence, for both queues $Q_{ij}^{k}(i)$ and $Q_{ij}^{k}(j)$ the following inequality holds

  $$\epsilon^3(1-\epsilon)\cdot\phi_{i,j}^{j,i} \leq (1-\epsilon^3)\cdot\phi_{ij,k}^{k,ij} + (1-\epsilon^3)\cdot\phi_{ij}^{k}.$$

  We set $\phi_{ij}^{k} = 0$, so inequality (13) becomes

  $$\epsilon^3(1-\epsilon)\cdot\phi_{i,j}^{j,i} \leq (1-\epsilon^3)\cdot\phi_{ij,k}^{k,ij}. \tag{29}$$

  Next, to satisfy (29), we set

  $$\phi_{ij,k}^{k,ij} = \epsilon^3(1-\epsilon)\cdot\phi_{i,j}^{j,i}/(1-\epsilon^3), \tag{30}$$

  where the second part of the inequality only depends on ϵ and λ, by substituting $\phi_{i,j}^{j,i}$ from (26). It follows that $\phi_{ij,k}^{k,ij} \geq 0$.
- Possible incoming flows to Q_i^{jk} are due to controls I_i,$I_{i,j}^{j,i}$,$I_{i,k}^{k,i}$,I_i^j ,I_i^k,$I_{ij,k}^{k,ij}$,$I_{ik,j}^{j,ik}$ and possible outgoing flows are due to controls $I_{jk,i}^{i,jk}$,$I_{i,j,k}^{jk,ik,jk}$,$I_{i,j}^{jk,ik}$,I_i^{jk}, where $i, j, k \in \{1, \ldots, 4\}$,$i \neq j \neq k$. For $Q_i^{jk}(i)$, inequality (13) becomes

  $$\begin{aligned}\epsilon^2(1-\epsilon)^2\cdot\left(\phi_i + \phi_{i,j}^{j,i} + \phi_{i,k}^{k,i}\right) + \epsilon^2(1-\epsilon)\cdot\left(\phi_i^j + \phi_i^k + \phi_{ij,k}^{k,ij} + \phi_{ik,j}^{j,ik}\right)\\ \leq (1-\epsilon^2)\cdot\left(\phi_{i,j,k}^{jk,ik,ij} + \phi_{jk,i}^{i,jk} + \phi_{i,j}^{jk,ik} + \phi_{i,k}^{jk,ij} + \phi_i^{jk}\right)\end{aligned}. \tag{31}$$

  For Q_j^{ik} and Q_k^{ij}, inequality (13) takes the form of (31), with the appropriate exchange of indices. Specifically, for Q_j^{ik} and Q_k^{ij}, we have the following inequalities, respectively

  $$\begin{aligned}\epsilon^2(1-\epsilon)^2\cdot\left(\phi_j + \phi_{i,j}^{j,i} + \phi_{j,k}^{k,j}\right) + \epsilon^2(1-\epsilon)\cdot\left(\phi_j^i + \phi_j^k + \phi_{ij,k}^{k,ij} + \phi_{jk,i}^{i,jk}\right)\\ \leq (1-\epsilon^2)\cdot\left(\phi_{i,j,k}^{jk,ik,ij} + \phi_{ik,j}^{j,ik} + \phi_{i,j}^{jk,ik} + \phi_{j,k}^{ik,ij} + \phi_j^{jk}\right)\end{aligned}, \tag{32}$$

  $$\begin{aligned}\epsilon^2(1-\epsilon)^2\cdot\left(\phi_k + \phi_{i,k}^{k,i} + \phi_{j,k}^{k,j}\right) + \epsilon^2(1-\epsilon)\cdot\left(\phi_k^i + \phi_k^j + \phi_{ik,j}^{j,ik} + \phi_{jk,i}^{i,jk}\right)\\ \leq (1-\epsilon^2)\cdot\left(\phi_{i,j,k}^{jk,ik,ij} + \phi_{ij,k}^{k,ij} + \phi_{i,k}^{jk,ij} + \phi_{j,k}^{ik,ij} + \phi_k^{ij}\right)\end{aligned}. \tag{33}$$

  All ϕ parameters in the left parts of inequalities (31), (32) and (33) have already been computed. Therefore, the unknown parameters at this point are $\phi_{i,j,k}^{jk,ik,ij}$, $\phi_{i,j}^{jk,ik}$, $\phi_{i,k}^{jk,ij}$, $\phi_{j,k}^{ik,ij}$, ϕ_i^{jk}, ϕ_j^{jk} and ϕ_k^{ij}. Assuming, $i \leq j \leq k$ we set

  $$\left.\begin{aligned}\phi_j^{ik} &= 0,\\ \phi_k^{ij} &= 0,\\ \phi_{i,k}^{jk,ij} &= 0,\\ \phi_{j,k}^{ik,ij} &= 0,\end{aligned}\right. \quad \text{if } i \leq j \leq k, \tag{34}$$

  and to satisfy (33) we set

  $$\begin{aligned}\phi_{i,j,k}^{jk,ik,ij} = \left[\epsilon^2(1-\epsilon)^2\left(\phi_k + \phi_{i,k}^{k,i} + \phi_{j,k}^{k,j}\right) + \epsilon^2(1-\epsilon)\left(\phi_k^i + \phi_k^j + \phi_{ik,j}^{j,ik} + \phi_{jk,i}^{i,jk}\right)\right]/(1-\epsilon^2)\\ -\phi_{ij,k}^{k,ij},\ i \leq j \leq k.\end{aligned} \tag{35}$$

Next, we substitute (35) into (32) and set

$$\begin{gathered}\phi_{i,j}^{jk,ik} = [\epsilon^2(1-\epsilon)^2\left(\phi_k + \phi_{i,k}^{k,i} + \phi_{j,k}^{k,j}\right) + \\ \epsilon^2(1-\epsilon)\cdot\left(\phi_k^i + \phi_k^j + \phi_{ik,j}^{j,ik} + \phi_{jk,i}^{i,jk}\right)]/(1-\epsilon^2) - \phi_{ik,j}^{j,ik} - \phi_{i,j,k}^{jk,ik,ij},\ i \le j \le k,\end{gathered} \tag{36}$$

to ensure that (32) is satisfied. Finally, substituting (36) into (24), we similarly obtain

$$\begin{gathered}\phi_i^{jk} = [\epsilon^2(1-\epsilon)^2\left(\phi_i + \phi_{i,j}^{j,i} + \phi_{i,k}^{k,i}\right) + \\ \epsilon^2(1-\epsilon)\left(\phi_i^j + \phi_i^k + \phi_{ij,k}^{k,ij} + \phi_{ik,j}^{j,ik}\right)]/(1-\epsilon^2) - \phi_{jk,i}^{i,jk} - \phi_{i,j,k}^{jk,ik,ij} - \phi_{i,j}^{jk,ik},\ i \le j \le k.\end{gathered} \tag{37}$$

We have to ensure that the quantities obtained in (35)-(37) are nonnegative. The formulas are getting very convoluted at this point but they are easily calculated as functions of the erasure probabilities and the arrival rates using symbolic computation packages. Using such a package (we used Maple 13.0), it is easy to see that the resulting expressions are nonnegative.

Example 14. By computing $\phi_{i,j}^{jk,ik}$ we find that $\phi_{i,j}^{jk,ik} = \epsilon^2\left(\lambda_2 - \lambda_3\right)\frac{\epsilon^2-\epsilon+1}{-\epsilon^6-\epsilon^5-\epsilon^4+\epsilon^2+\epsilon+1}$. From (19) we have $\lambda_2 - \lambda_3 \ge 0$. Additionally, $-\epsilon + 1 > 0$, so the numerator is nonnegative. Also, the denominator is positive, because $-\epsilon^6 - \epsilon^5 - \epsilon^4 + \epsilon^2 + \epsilon + 1 = \left(1+\epsilon+\epsilon^2\right)\left(1-\epsilon^4\right)$, where both parentheses are positive. Therefore, $\phi_{i,j}^{jk,ik} \ge 0$.

Level 4 At level 4, there are 4 virtual queues of the form Q_{ijk}^l (which corresponds to real queues $Q_{ijk}^l\,(i)$, $Q_{ijk}^l\,(j)$, $Q_{ijk}^l\,(k)$), 6 virtual queues of the form Q_{ij}^{kl} (which corresponds to real queues $Q_{ij}^{kl}\,(i)$,$Q_{ij}^{kl}\,(j)$) and 4 queues of the form Q_i^{jkl} (corresponding to real queue $Q_i^{jkl}\,(i)$).

- Incoming flows to queue Q_{ijk}^l are due to controls $I_{ij,k}^{k,ij}$, $I_{ik,j}^{j,ik}$, $I_{jk,i}^{i,jk}$ and $I_{i,j,k}^{jk,ik,ij}$, with probability $\epsilon^3(1-\epsilon)$, while outgoing flows are due to controls $I_{ijk,l}^{l,ijk}$ and I_{ijk}^l with probability $1-\epsilon^3$. We set $\phi_{ijk}^l = 0$ so that inequality (13) becomes

$$\epsilon^3(1-\epsilon)\left(\phi_{ij,k}^{k,ij} + \phi_{ik,j}^{j,ik} + \phi_{jk,i}^{i,jk} + \phi_{i,j,k}^{jk,ik,ij}\right) \le \left(1-\epsilon^3\right)\phi_{ijk,l}^{l,ijk}. \tag{38}$$

To satisfy (38), we set

$$\phi_{ijk,l}^{l,ijk} = \epsilon^3(1-\epsilon)\left(\phi_{ij,k}^{k,ij} + \phi_{ik,j}^{j,ik} + \phi_{jk,i}^{i,jk} + \phi_{i,j,k}^{jk,ik,ij}\right)/\left(1-\epsilon^3\right). \tag{39}$$

- Incoming flows to queue Q_{ij}^{kl} are due to controls $I_{i,j}^{j,i}$, $I_{ij,k}^{k,ij}$, $I_{ij,l}^{l,ij}$, $I_{ik,j}^{j,ik}$, $I_{il,j}^{j,il}$, $I_{jk,i}^{i,jk}$, $I_{jl,i}^{i,jl}$, $I_{i,j,k}^{jk,ik,ij}$, $I_{i,j,l}^{jl,il,ij}$, with probability $\epsilon^2(1-\epsilon)^2$, and $I_{i,j}^{jk,ik}$, $I_{i,j}^{jl,il}$, $I_{ijk,l}^{l,ijk}$, $I_{ijl,k}^{k,ijl}$ with probability $\epsilon^2(1-\epsilon)$. Outgoing flows are due to controls $I_{ij,kl}^{kl,ij}$, $I_{ij,k,l}^{kl,ijl,ijk}$ and I_{ij}^{kl} with probability $1-\epsilon^2$. Therefore, inequality (13) becomes

$$\begin{gathered}\epsilon^2(1-\epsilon)^2\left(\phi_{i,j}^{j,i} + \phi_{ij,k}^{k,ij} + \phi_{ij,l}^{l,ij} + \phi_{ik,j}^{j,ik} + \phi_{il,j}^{j,il} + \phi_{jk,i}^{i,jk} + \phi_{jl,i}^{i,jl} + \phi_{i,j,k}^{jk,ik,ij} + \phi_{i,j,l}^{jl,il,ij}\right) \\ +\epsilon^2(1-\epsilon)\left(\phi_{i,j}^{jk,ik} + \phi_{i,j}^{jl,il} + \phi_{ijk,l}^{l,ijk} + \phi_{ijl,k}^{k,ijl}\right) \le \left(1-\epsilon^2\right)\left(\phi_{ij,kl}^{kl,ij} + \phi_{ij,k,l}^{kl,ijl,ijk} + \phi_{ij}^{kl}\right).\end{gathered} \tag{40}$$

We set

$$\phi_{ij}^{kl} = 0, \tag{41}$$

so that (40) becomes now

$$\begin{gathered}\epsilon^2(1-\epsilon)^2\left(\phi_{i,j}^{j,i} + \phi_{ij,k}^{k,ij} + \phi_{ij,l}^{l,ij} + \phi_{ik,j}^{j,ik} + \phi_{il,j}^{j,il} + \phi_{jk,i}^{i,jk} + \phi_{jl,i}^{i,jl} + \phi_{i,j,k}^{jk,ik,ij} + \phi_{i,j,l}^{jl,il,ij}\right) \\ +\epsilon^2(1-\epsilon)\left(\phi_{i,j}^{jk,ik} + \phi_{i,j}^{jl,il} + \phi_{ijk,l}^{l,ijk} + \phi_{ijl,k}^{k,ijl}\right) \le \left(1-\epsilon^2\right)\left(\phi_{ij,kl}^{kl,ij} + \phi_{ij,k,l}^{kl,ijl,ijk}\right).\end{gathered} \tag{42}$$

Similarly, for queue Q_{kl}^{ij}, inequality (13) becomes

$$\begin{gathered}\epsilon^2(1-\epsilon)^2\left(\phi_{k,l}^{l,k} + \phi_{kl,i}^{i,kl} + \phi_{kl,j}^{j,kl} + \phi_{ik,l}^{l,ik} + \phi_{jk,l}^{l,jk} + \phi_{il,k}^{k,il} + \phi_{jl,k}^{k,jl} + \phi_{i,k,l}^{kl,il,ik} + \phi_{j,k,l}^{kl,jl,jk}\right) + \\ \epsilon^2(1-\epsilon)\left(\phi_{k,l}^{il,ik} + \phi_{k,l}^{jl,jk} + \phi_{ikl,j}^{j,ikl} + \phi_{jkl,i}^{i,jkl}\right) \le \left(1-\epsilon^2\right)\left(\phi_{ij,kl}^{kl,ij} + \phi_{kl,i,j}^{ij,jkl,ikl}\right).\end{gathered} \tag{43}$$

All ϕ parameters in the left parts of inequalities (40) and (43) have already been computed. Therefore, there are two inequalities with 3 unknown parameters: $\phi_{ij,kl}^{kl,ij}$, $\phi_{ij,k,l}^{kl,ijl,ijk}$ and $\phi_{kl,i,j}^{ij,jkl,ikl}$. Assume for the rest of this case that $i \le j \le k \le l$. We then set

$$\phi_{kl,i,j}^{ij,jkl,ikl} = 0. \tag{44}$$

Now, there are only 2 unknown parameters, so to satisfy (43) we set

$$\begin{aligned} \phi_{ij,kl}^{kl,ij} = \; & [\epsilon^2(1-\epsilon)^2 \left(\phi_{k,l}^{l,k} + \phi_{kl,i}^{i,kl} + \phi_{kl,j}^{j,kl} + \phi_{ik,l}^{l,ik} + \phi_{jk,l}^{l,jk} + \phi_{il,k}^{k,il} + \phi_{jl,k}^{k,jl} + \phi_{i,k,l}^{kl,il,ik} + \phi_{j,k,l}^{kl,jl,jk} \right) \\ & + \epsilon^2(1-\epsilon) \left(\phi_{k,l}^{il,ik} + \phi_{k,l}^{jl,jk} + \phi_{ikl,j}^{j,ikl} + \phi_{jkl,i}^{i,jkl} \right)] / \left(1-\epsilon^2\right) \end{aligned} \tag{45}$$

and to satisfy (40) we set

$$\begin{aligned} \phi_{ij,k,l}^{kl,ijl,ijk} = \; & [\epsilon^2(1-\epsilon)^2 \left(\phi_{i,j}^{j,i} + \phi_{ij,k}^{k,ij} + \phi_{ij,l}^{l,ij} + \phi_{ik,j}^{j,ik} + \phi_{il,j}^{j,il} + \phi_{jk,i}^{i,jk} + \phi_{jl,i}^{i,jl} + \phi_{i,j,k}^{jk,ik,ij} + \phi_{i,j,l}^{jl,il,ij} \right) \\ & + \epsilon^2(1-\epsilon) \left(\phi_{i,j}^{jk,ik} + \phi_{i,j}^{jl,il} + \phi_{ijk,l}^{l,ijk} + \phi_{ijl,k}^{k,ijl} \right)] / \left(1-\epsilon^2\right) - \phi_{ij,kl}^{kl,ij} \end{aligned} . \tag{46}$$

- For queues of the form Q_i^{jkl}, incoming flows are due to controls of the form $I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m}$, $i \in \mathcal{D}_1,\ |\mathcal{D}_j \cup \mathcal{L}_j| \le 3,\ j \in \{1,\ldots,3\}$, as well as controls of the form $I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m}$, $i \in \mathcal{D}_1,\ |\mathcal{D}_1| \ge 2,\ |\mathcal{D}_j \cup \mathcal{L}_j| = 4,\ j \in \{1,\ldots,3\}$. Outgoing flows are due to controls of the form $I_{\mathcal{D}_1,\ldots,\mathcal{D}_m}^{\mathcal{L}_1,\ldots,\mathcal{L}_m}$, $i \in \mathcal{D}_1,\ |\mathcal{D}_1| = 1,\ |\mathcal{D}_j \cup \mathcal{L}_j| = 4,\ j \in \{1,\ldots,4\}$, with probability $(1-\epsilon)$. Therefore, inequality (13) becomes

$$\begin{gathered} \epsilon\left(1-\epsilon\right)^3 \left(\phi_i + \sum_{j \neq i} \phi_{i,j}^{j,i} + \sum_{j,k \neq i} \phi_{jk,i}^{i,jk} + \sum_{j,k \neq i} \phi_{ij,k}^{k,ij} + \sum_{j,k \neq i} \phi_{i,j,k}^{jk,ik,ij} \right) + \\ \epsilon\left(1-\epsilon\right)^2 \left(\sum_{j \neq i} \phi_i^j + \sum_{j,k \neq i} \phi_{i,j}^{jk,ik} + \sum_{j,k,l \neq i} \phi_{ijk,l}^{l,ijk} \right) + \\ \epsilon\left(1-\epsilon\right) \left(\sum_{j,k \neq i} \phi_i^{jk} + \sum_{j,k,l \neq i} \phi_{ij,kl}^{kl,ij} + \sum_{j,k,l \neq i} \phi_{ij,k,l}^{kl,ijl,ijk} \right) \le \\ \left(1-\epsilon\right) (\phi_{i,j,k,l}^{jkl,ikl,ijl,ijk} + \phi_{jkl,i}^{i,jkl} + \phi_{jk,i,l}^{il,jkl,ijk} + \phi_{i,j,k}^{jkl,ikl,ijl} + \\ + \phi_{i,j,l}^{jkl,ikl,ijk} + \phi_{i,k,l}^{jkl,ijl,ijk} + \phi_{i,j}^{jkl,ikl} + \phi_{i,k}^{jkl,ijl} + \phi_{i,l}^{jkl,ikl} + \phi_i^{jkl}). \end{gathered} \tag{47}$$

  Similar inequalities can be formed for Q_j^{ikl}, Q_k^{ijl} and Q_l^{ijk}. In general, if $i \le j \le k \le l$, we set

$$\begin{aligned} \phi_{a,b,c}^{bcd,acd,abd} &= 0,\ \forall\left(a,b,c\right) \neq \left(i,j,k\right),\ a,b,c,d \in \{i,j,k,l\} \\ \phi_{a,b}^{bcd,acd} &= 0,\ \forall\left(a,b\right) \neq \left(i,j\right),\ a,b,c,d \in \{i,j,k,l\} \\ \phi_a^{bcd} &= 0,\ \forall a \neq i,\ a,b,c,d \in \{i,j,k,l\}. \end{aligned} \tag{48}$$

  Therefore, in the inequality concerning Q_l^{ijk} the only unknown parameter is $\phi_{i,j,k,l}^{jkl,ikl,ijl,ijk}$. We convert this inequality to equality and set $\phi_{i,j,k,l}^{jkl,ikl,ijl,ijk}$ to the value satisfying it. Then, we proceed to the inequality concerning Q_k^{ijl}, where we substitute $\phi_{i,j,k,l}^{jkl,ikl,ijl,ijk}$ and set a value for the remaining unknown, $\phi_{i,j,k}^{jkl,ikl,ijl}$, such that it satisfies the respective equality. Then, in the inequality concerning Q_j^{ikl} we substitute $\phi_{i,j,k,l}^{jkl,ikl,ijl,ijk}$ and $\phi_{i,j,k}^{jkl,ikl,ijl}$, and set $\phi_{i,j}^{jkl,ikl}$ to the value satisfying the respective equality. Finally, we substitute $\phi_{i,j,k,l}^{jkl,ikl,ijl,ijk}$, $\phi_{i,j,k}^{jkl,ikl,ijl}$ and $\phi_{i,j}^{jkl,ikl}$ to (47) and set ϕ_i^{jkl} in a similar way. Again, it is easy to verify through symbolic computation that all computed quantities are nonnegative.

Finally, to ensure that (16) is satisfied, we calculate the sum of all flows, and find

$$\sum_{I \in \mathcal{I}} \phi_I = \sum_{i=1}^{4} \frac{\lambda_i}{1-\epsilon^i}.$$

Since, by assumption, it holds $\sum_{i=1}^4 \frac{\lambda_i}{1-\epsilon^i} \le 1$, we conclude that $\sum_{I \in \mathcal{I}} \phi_I \le 1$, as desired. Hence, we have proved the following result.

Theorem 15. *For the case of 4 users, and for i.i.d erasure events, the stability region of the system is given by*

$$\mathcal{R}_{\Pi} = \left\{ \boldsymbol{\lambda} : \max_{\sigma \in \mathcal{P}} \sum_{i=1}^{4} \frac{\lambda_{\sigma(i)}}{1-\epsilon^{i}} \leq 1 \right\}$$

Moreover, the policy $\pi^* \in \Pi$ described in Section IV-B using the XOR controls described in Table I is stabilizing. The stability region coincides with the information theoretic capacity region of the standard broadcast erasure channel with feedback, and is within one bit (actually, $O(2^{-L/A^*})$ bits according to Theorem 12, where $A^* = \max_{\sigma} A_{\sigma}$) from the capacity of the "extended" broadcast erasure channel with feedback. The latter is equal to the stability region of the system under any coding strategy.

VII. Implementation Issues

A. Packet overhead

As mentioned in Section IV, for the proposed network coding scheme to work, every user must know the identities of all native packets that constitute a composite (i.e. XOR combination) packet it receives. Having this information, a user is able to decode the native packet destined for it. A simple mechanism that can be used to provide users with this information is equipping every native packet with a Packet ID, which consists of the packet's destination and a sequence number. If a transmitted packet is composed of m packets, then it contains in its packet header the m packet IDs.

To compute the overhead bits needed to implement the above mechanism, we need to find the maximum number of Packet IDs that may be included in a packet that is placed in a virtual queue of a certain level. This is expressed in Lemma 16 below. In the following, when we say that a packet *comes from level* k (or *exits level* k) we mean that it is an XOR combination of packets placed in queues of levels 1 to k (with at least one packet being in a level k queue).

Lemma 16. *Under the coding scheme of Section III, it holds*

a) Any packet placed in queues at sublevel $k.n$, $n = 1, 2, \ldots, k-1$, $k \geq 2$, contains at most $(k-1)!$ packet IDs.

b) Any packet exiting level $k \geq 2$ contains at most $k!$ packet IDs.

Proof: We use induction on k to prove the Lemma. For $k = 2$, the Lemma follows immediately from the Rules for Packet Movement in Section III. We now assume that the Lemma holds for levels 2 up to $k-1$ and show that it also holds for level k. We first prove part a) of the Lemma by induction on n.

Part a): If a packet p is placed in a queue at the lowest sublevel of level k, i.e. $k.1$, then according to Lemma 7, p comes from levels $l \leq k-1$. Hence, according to part b) of the inductive hypothesis, it contains at most $(k-1)!$ packet IDs, so that part a) holds for $n = 1$. Assume next that part a) holds for all packets p placed at any sublevel from $k.1$ up to $k.n$ with $2 \leq n < k-1$, i.e. assume that all packets p in sublevels from $k.1$ up to $k.n$ contain at most $(k-1)!$ packet IDs. We shall prove that any packet in sublevel $k.(n+1)$ also contains at most $(k-1)!$ packet IDs. According to Lemma 7 for a packet p at sublevel $k.(n+1)$, one of the following two cases holds.

1. Packet p comes from level l, where $2 \leq l \leq k-1$. Then, according to part b) of the inductive hypothesis, p contains at most $(k-1)!$ packet IDs.

2. Packet p was placed before the current slot transmission at a queue in a lower sublevel of the same level, i.e. a sublevel from $k.1$ up to $k.n$. According to the inductive hypothesis on n, packets in these sublevels contain at most $(k-1)!$ packet IDs. Since Lemma 7 states that packets from lower sublevels are merely copied to higher sublevels, it follows that the maximum number of packet IDs they contain remains the same, so packet p at sublevel $k.(n+1)$ will also contain at most $(k-1)!$ packet IDs. Therefore, packets at all sublevels $k.n$, $n = 1, 2, \ldots, k-1$, $k \geq 2$, contain at most $(k-1)!$ packet IDs. This completes the proof of part a) of the Lemma.

To prove part b) of the Lemma, consider a packet p exiting level k. This packet is of the form $p = p_{\mathcal{D}_1}^{\mathcal{L}_1} \oplus \ldots \oplus p_{\mathcal{D}_m}^{\mathcal{L}_m}$, where each $p_{\mathcal{D}_r}^{\mathcal{L}_r}$ belongs to a queue of at most level k, hence the maximum number of packet IDs p may contain is the sum of the packet IDs contained in packets $p_{\mathcal{D}_r}^{\mathcal{L}_r}$, $r \in \{1, \ldots, m\}$, which is at most $m(k-1)!$, due to part a). From Lemma 6, it holds $m \leq k$, therefore any packet exiting level k contains at most $k(k-1)! = k!$ packet IDs. ■

Up to level 4, the maximum number of Packet IDs that may need to be included in a packet is $4! = 24$. Assuming a packet ID of 20 bits and packet lenght of 12000 bits, the overhead is approximately 4%. It can be seen that the maximum number of Packet IDs needed increases dramatically with the number of users N and it is very important to address this matter as N increases. As a future work, various suboptimal policies that reduce the necessary number of Packet IDs can be investigated. For example, the transmitter may choose not to send packet combinations if the resulting packet header exceeds a certain number of Packet IDs. Another policy towards this direction could involve coding of packets only until a certain level. Specifically, for N users, only the virtual queues until level l could be created, where $l < N$. In case a packet is received by more than l users, additional receivers would be ignored and the packet would be placed in one of the level l queues.

B. *Queue stability at the receivers*

As mentioned in Section IV, another problem that may arise is possible instability of queues at the receivers, where all packets received by a certain user are stored. A simple way to avert this possibility is to take advantage of the fact that when the queue sizes at the based station become empty, all packets formed during previous transmissions are not needed at the receivers. Therefore, we can let the base station inform all receivers when its queues become emtpy, by, for example, leaving a slot empty after a series of transmissions taking place when the queues are nonempty. Under this modification, using standard results from regenerative theory, it can be shown that the system is stable if and only if the total queue size at the base station is stable.

VIII. Conclusions

In this work, we presented a network coding scheme for the broadcast erasure channel with N multiple unicast sessions based on the coding scheme we proposed in [4]. This scheme is characterized by low complexity, as only XOR operations are allowed. Also, instant decodability, i.e. the ability of any user that receives a coded packet to instantly decode its own native packet, is ensured. While in [4] the possible transitions of packets between the virtual queues of the coding scheme were determined only for the case of 3 users, in this paper we give general rules for packet movement.

Furthermore, we assumed random packet arrivals and presented a stabilizing policy based on this coding scheme. We then derived a capacity outer bound for the system under examination. For the case of 4 users and i.i.d. erasure events, we proved that any rate within the capacity outer bound is stable under the proposed policy, therefore the stability region of the system is identical to the capacity outer bound of the BEC channel with feedback.

Finally, implementation issues were examined, such as the increase of packet overhead as the number of users increases, which is due to the number of packet addresses needed to completely describe a coded packet. The exact number of addresses needed in the general case of N users was found to be $N!$. Future work could be aimed towards the development of suboptimal variations of the proposed policy that will require a smaller number of packet addresses,this reducing packet overhead.

Appendix

A. *Proof Of Theorem 13*

We need some preliminary definitions. Define the sets $\mathcal{N}_1 = \emptyset$ and $\mathcal{N}_i = \{1, 2, \ldots, i-1\}$, for $i \in \mathcal{N}$ with $i \geq 2$, as well as

$$\mathcal{Q}_i = \left\{ Q_{\mathcal{D}}^{\mathcal{L}}(i) : \ i \in \mathcal{D}, \text{ and } \mathcal{L}, \mathcal{D} - \{i\} \subseteq \mathcal{N}_i \right\}$$

$$\mathcal{I}_i = \left\{ I_{\mathcal{D}, \mathcal{D}_2, \ldots, \mathcal{D}_m}^{\mathcal{L}, \mathcal{L}_2, \ldots, \mathcal{L}_m} : \ i \in \mathcal{D}, \text{ and } \mathcal{L}, \mathcal{D} - \{i\} \subseteq \mathcal{N}_i \right\}.$$

Notice that $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset$ for $i \neq j$. This is due to the fact that the existence of a control $I \in \mathcal{I}_i \cap \mathcal{I}_j$ would imply that $i \in \mathcal{N}_j$ as well as $j \in \mathcal{N}_i$, which is impossible. We also define the set $\mathcal{Q}_i^{\mathcal{N}}$ in the subnetwork consisting of queues (i.e. each node is a queue, as described in Section IV) as follows:

$$\mathcal{Q}_i^{\mathcal{N}} = \left\{ Q_{\mathcal{D}}^{\mathcal{L}}(i) : \ i \in \mathcal{D}, \text{ and } \mathcal{D}, \mathcal{L} \subseteq \mathcal{N} \right\} \cup \{d\},$$

Denote with $\mathcal{C}_{out}(\mathcal{Q}_i)$ the set of all outgoing links in the cut $\left[\mathcal{Q}_i, \mathcal{Q}_i^{\mathcal{N}} - \mathcal{Q}_i\right]$, i.e.

$$\mathcal{C}_{out}(\mathcal{Q}_i) = \left\{ e = (i,j) \in \mathcal{E} : \ i \in \mathcal{Q}_i, \ j \in \mathcal{Q}_i^{\mathcal{N}} - \mathcal{Q}_i \right\}$$

and notice that, according to the Rules for Packet Movement, the set of all incoming links to the cut is empty, i.e.,

$$\mathcal{C}_{in}\left(\mathcal{Q}_i\right) = \left\{e = (i,j) \in \mathcal{E} :\ i \in \mathcal{Q}_i^{\mathcal{N}} - \mathcal{Q}_i,\ j \in \mathcal{Q}_i\right\} = \emptyset \tag{49}$$

To prove Theorem 13, it suffices to show that, under (12) and (14)-(16), it holds

$$\sum_{i=1}^{\mathcal{N}} \frac{\lambda_i}{1 - \epsilon_{\mathcal{N}-\mathcal{N}_i}} \leq 1, \tag{50}$$

since the same argument can be repeated verbatim for any permutation $\sigma(i)$, $i \in \mathcal{N}$. Summing (12) over all $m \in \mathcal{Q}_i$ and using (17) yields

$$\sum_{I \in \mathcal{I}_i} \phi_I \sum_{m \in \mathcal{Q}_i} \sum_{e=(l,m) \in \mathcal{E}_{in}^m} \mu_l\left(I\right) p_e^l(I) + \lambda_i \leq \sum_{I \in \mathcal{I}_i} \sum_{m \in \mathcal{Q}_i} \sum_{e \in \mathcal{E}_{out}^m} p_e^m(I) \mu_m\left(I\right) \phi_I,\ \forall\, i \in \mathcal{N},$$

or, rearranging the terms,

$$\lambda_i \leq \sum_{I \in \mathcal{I}_i} \left(\sum_{m \in \mathcal{Q}_i} \left(\sum_{e \in \mathcal{E}_{out}^m} p_e^m(I) \mu_m\left(I\right) - \sum_{e=(l,m) \in \mathcal{E}_{in}^m} \mu_l\left(I\right) p_e^l(I) \right) \right) \phi_I. \tag{51}$$

But (49) and the construction of $\mathcal{C}_{out}(\mathcal{Q}_i)$, $\mathcal{C}_{in}(\mathcal{Q}_i)$ imply

$$\begin{aligned} &\sum_{m \in \mathcal{Q}_i} \left(\sum_{e \in \mathcal{E}_{out}^m} p_e^m(I) \mu_m\left(I\right) - \sum_{e=(l,m) \in \mathcal{E}_{in}^m} \mu_l\left(I\right) p_e^l(I) \right) \\ &= \sum_{e=(l,m) \in \mathcal{C}_o(\mathcal{Q}_i)} \mu_l\left(I\right) p_e^l(I) - \sum_{e=(l,m) \in \mathcal{C}_{in}(\mathcal{Q}_i)} \mu_l\left(I\right) p_e^l(I) = \sum_{e=(l,m) \in \mathcal{C}_o(\mathcal{Q}_i)} \mu_l\left(I\right) p_e^l(I). \end{aligned}$$

Under any control $I_{\mathcal{D},\mathcal{D}_2,\ldots,\mathcal{D}_m}^{\mathcal{L},\mathcal{L}_2,\ldots,\mathcal{L}_m} \in \mathcal{I}_i$, it holds

$$\sum_{e=(l,m) \in \mathcal{C}_o(\mathcal{Q}_i)} \mu_l\left(I\right) p_e^l(I) = 1 - \epsilon_{\mathcal{N}-\mathcal{N}_i}. \tag{52}$$

This follows from the fact that, when $I_{\mathcal{D},\mathcal{D}_2,\ldots,\mathcal{D}_m}^{\mathcal{L},\mathcal{L}_2,\ldots,\mathcal{L}_m}$ is applied, it holds $\mu_l\left(I\right) = 1$ for $l = Q_{\mathcal{D}}^{\mathcal{L}}$ and $\mu_l\left(I\right) = 0$ for all other queues in $\mathcal{Q}_i^{\mathcal{N}}$, according to (9). Moreover, under $I_{\mathcal{D},\mathcal{D}_2,\ldots,\mathcal{D}_m}^{\mathcal{L},\mathcal{L}_2,\ldots,\mathcal{L}_m}$, a native packet for user i is transferred to one of the queues in $\mathcal{Q}_i^{\mathcal{N}} - \mathcal{Q}_i$ iff the packet is not erased by all users in $\mathcal{N} - \mathcal{N}_i$.

Hence, (51) yields through (52)

$$\lambda_i \leq \sum_{I \in \mathcal{I}_i} \left(1 - \epsilon_{\mathcal{N}-\mathcal{N}_i}\right) \phi_I$$

and, summing over all $i \in \mathcal{N}$, we conclude that

$$\sum_{i \in \mathcal{N}} \frac{\lambda_i}{1 - \epsilon_{\mathcal{N}-\mathcal{N}_i}} \leq \sum_{i \in \mathcal{N}} \sum_{I \in \mathcal{I}_i} \phi_I$$

However, since $\mathcal{I}_i \cap \mathcal{I}_j = \emptyset$ for all $i \neq j$, it holds $\sum_{i \in \mathcal{N}} \sum_{I \in \mathcal{I}_i} \phi_I \leq \sum_{I \in \mathcal{I}} \phi_I \leq 1$ and (50) is proved.

B. *Proof Of Theorem 12*

We first need to establish some notation and prove a few intermediate results. We consider the "extended" broadcast erasure channel (BEC), where the transmitter has the option of not transmitting in a given slot (as opposed to the "standard" BEC that appears in the literature). This is equivalent to considering that the transmitter sends in this slot a special (null) symbol, denoted as $\varnothing$. Hence, in information theoretic terms, given a standard point-to-point BEC with an input alphabet of $\mathcal{X}$ and output alphabet of $\mathcal{Y} = \mathcal{X} \cup \{*\}$, where $*$ denotes an erasure, the extended point-to-point BEC has input alphabet $\mathcal{X}' = \mathcal{X} \cup \{\varnothing\}$ and output alphabet $\mathcal{Y}' = \mathcal{X}' \cup \{*\} = \mathcal{X} \cup \{*, \varnothing\}$. Since we consider feedback, we assume that, if the transmitter sends symbol $\varnothing$, *all* users send $\varnothing$ as feedback back to the transmitter. Hence, at slot l, each user can send feedback $Z \in \{ACK, NACK, \varnothing\}$ to the transmitter, where

ACK (resp. $NACK$) denotes a successfull reception (resp. erasure) of a non-null symbol, while $\varnothing$ denotes a null symbol transmission (and reception).

The N user version of the extended BEC follows from a simple "vectorization" procedure. Specifically, let $\mathcal{N} = \{1, \dots, N\}$ be the set of N users and denote with W_i the message for user $i \in \mathcal{N}$. The transmitted symbol at slot l is denoted as $X(l)$ (with $X(l) \in \mathcal{X}'$) and we also introduce the shortcut notation $X^l \triangleq (X(1), \dots, X(l))$. Furthermore, let $Y_i(l) \in \mathcal{Y}'$ be the symbol received by user i at slot l, while $Z_i(l) \in \{ACK, NACK, \varnothing\}$ is the feedback sent by user i to the transmitter at slot l. We can also define an auxiliary random variable $\hat{Z}_i(l) \in \{ACK, NACK\}$ that is independent of $X(l)$ and all previously generated random variables (up to slot l) so that it holds

$$Z_i(l) = \begin{cases} \hat{Z}_i(l) & \text{if } X(l) \neq \varnothing \\ \varnothing & \text{if } X(l) = \varnothing \end{cases}$$

Notice that, for any $z \neq \varnothing$, the events $\{Z_i(l) = z\}$ and $\{\hat{Z}_i(l) = z, F(l) = 1\}$ are identical. We now introduce the following "vectorized" entities

$$\begin{gathered} W_{[1,j]} = (W_1, \dots, W_j) \\ Y_i^l = (Y_i(1), \dots, Y_i(l)) \\ \boldsymbol{Y}_{[1,j]}(l) = (Y_1(l), \dots, Y_j(l)), \quad \boldsymbol{Y}_{[1,j]}^l = (\boldsymbol{Y}_{[1,j]}(1), \dots, \boldsymbol{Y}_{[1,j]}(l)) \\ \boldsymbol{Z}_{[1,j]}(l) = (Z_1(l), \dots, Z_j(l)), \quad \boldsymbol{Z}_{[1,j]}^l = (\boldsymbol{Z}_{[1,j]}(1), \dots, \boldsymbol{Z}_{[1,j]}(l)) \\ \hat{\boldsymbol{Z}}_{[1,j]}(l) = (\hat{\boldsymbol{Z}}_1(l), \dots, \hat{\boldsymbol{Z}}_j(l)) \end{gathered}$$

and use the shortcut $\boldsymbol{Y} = \boldsymbol{Y}_{[1,N]}$, $\boldsymbol{Y}^l = \boldsymbol{Y}_{[1,N]}^l$ (with similar interpretation for $\boldsymbol{Z}$, $\boldsymbol{Z}^l$).

The subsequent analysis closely follows the approach in [10], with some necessary variations due to the fact that $\boldsymbol{Z}(l)$ are $X(l)$ are not independent. The following Lemma can be proved by straightforward manipulations of information measures.

Lemma 17. *Let A, B, C, D be discrete random variables. The following identities hold.*

1) *Conditioning can be added to either part of mutual information:*

$$I(A; B|C, D) = I(A, C; B|C, D) = I(A; B, C|C, D) = I(A, C; B, C|C, D)$$

2) *Let B be independent of the joint ensemble (C, D). It then holds $I(A, B; C|D) = I(A; C|B, D)$.*
3) *Let D be independent of the joint ensemble (A, B, C). It then holds $I(A; B|C, D) = I(A; B|C)$.*
4) *Conditioning can be augmented by redundant condition, i.e. if the event $\{B = b\}$ implies $\{C = c_b\}$, it then holds $H(A|B, D) = H(A|B, C, D)$.*
5) *It holds $I(A; B|C) = I(A; B|C, D) + I(A; D|C) - I(A; D|B, C)$.*

We now consider an arbitrary code $\mathfrak{C}$ for the extended BEC with feedback (see [8] for a detailed description of encoding and decoding functions of $\mathfrak{C}$) and denote $\pi(l) = \Pr(X(l) \neq \varnothing)$ and $F(l) = \mathbb{I}[X(l) \neq \varnothing]$. The following results, whose proofs can be found, respectively, in sections C, D of the Appendix, will be used.

Lemma 18. *For any rate $\boldsymbol{R} = (R_1, \dots, R_N)$ that is achievable under $\mathfrak{C}$, and for any $j \in \mathcal{N}$, it holds*

$$n \sum_{k=1}^{j} R_k \leq \sum_{l=1}^{n} \Big[h(\pi(l)) + (1 - \pi(l))(1 - \epsilon_{\{1,\dots,j\}}) I(W_{[1,j]}; X(l) | \boldsymbol{Y}_{[1,j]}^{l-1}, \boldsymbol{Z}^{l-1}, F(l) = 1) \Big] + o(n)$$

where $h(\cdot)$ is Shannon's entropy function.

Lemma 19. *For any rate $\boldsymbol{R} = (R_1, \dots, R_N)$ that is achievable under $\mathfrak{C}$, and for any $j \in \mathcal{N}$, it holds*

$$n \sum_{k=1}^{j} R_k \geq (1 - \epsilon_{\{1,\dots,j+1\}}) \sum_{l=1}^{n} (1 - \pi(l)) I(W_{[1,j]}; X(l) | \boldsymbol{Y}_{[1,j+1]}^{l-1}, \boldsymbol{Z}^{l-1}, F(l) = 1)$$

Applying Lemma 18 for $j-1$ yields

$$\frac{n\sum_{k=1}^{j-1}R_k}{1-\epsilon_{\{1,\dots,j-1\}}} \leq o(n) + \sum_{l=1}^{n}\frac{h(\pi(l))}{1-\epsilon_{\{1,\dots,j-1\}}} + \sum_{l=1}^{n}(1-\pi(l))I(W_{[1,j-1]};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1) \tag{53}$$

where the second line was produced by using the inequality

$$\begin{aligned}(1-\pi(l))I(W_{[1,j-1]};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1) &= I(W_{[1,j-1]};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l))\\ &\overset{it.5}{\leq} I(W_{[1,j-1]};X(l)|\boldsymbol{Y}_{[1,j]}^{l-1},\boldsymbol{Z}^{l-1},F(l)) + I(Y_j^{l-1};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l))\\ &= (1-\pi(l))\left[I(W_{[1,j-1]};X(l)|\boldsymbol{Y}_{[1,j]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1) + I(Y_j^{l-1};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1)\right]\end{aligned} \tag{54}$$

and applying Lemma 19, for $j-1$, to the first term in the last line of (54). Hence, we arrive at

$$\begin{aligned}n\sum_{k=1}^{j-1}R_k\left(\frac{1}{1-\epsilon_{\{1,\dots,j-1\}}}-\frac{1}{1-\epsilon_{\{1,\dots,j\}}}\right) &\leq o(n) + \frac{1}{1-\epsilon_{\{1,\dots,j-1\}}}\sum_{l=1}^{n}h(\pi(l))\\ &\quad + \sum_{l=1}^{n}(1-\pi(l))I(Y_j^{l-1};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1)\end{aligned} \tag{55}$$

We are now ready to prove Theorem 12. We only consider the identity permutation (i.e. $\sigma(i)=i$), since all other permutations are handled similarly. Summing (55) for $j=2,\dots,N$, applying Lemma 18 for $j=N$ and summing the results yields after some manipulations (which involve a change of order summation between j and k)

$$\begin{aligned}n\sum_{k=1}^{N}\frac{R_k}{1-\epsilon_{\{1,\dots,k\}}} &\leq \left(\sum_{j=1}^{N}\frac{1}{1-\epsilon_{\{1,\dots,j\}}}\right)\sum_{l=1}^{n}h(\pi(l)) + \sum_{l=1}^{n}(1-\pi(l))\sum_{j=2}^{N}I(Y_j^{l-1};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1)\\ &\quad + \sum_{l=1}^{n}(1-\pi(l))I(W_{[1,N]};X(l)|\boldsymbol{Y}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1) + o(n)\end{aligned} \tag{56}$$

For notational compactness, we hereafter denote $A=\sum_{j=1}^{N}\frac{1}{1-\epsilon_{\{1,\dots,j\}}}$. It also holds

$$\begin{aligned}L \geq H(X(l)|F(l)=1) &= I(X(l);\boldsymbol{Y}_{[1,N]}^{l-1},\boldsymbol{Z}^{l-1}|F(l)=1) + H(X(l)|\boldsymbol{Y}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1)\\ &= \sum_{j=2}^{N}I(Y_j^{l-1};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1) + H(X(l)|\boldsymbol{Y}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1)\\ &\geq \sum_{j=2}^{N}I(Y_j^{l-1};X(l)|\boldsymbol{Y}_{[1,j-1]}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1) + H(W_{[1,N]};X(l)|\boldsymbol{Y}^{l-1},\boldsymbol{Z}^{l-1},F(l)=1)\end{aligned} \tag{57}$$

where the second line is derived by applying the chain rule over j. Inserting (57) into (56) yields

$$n\sum_{k=1}^{N}\frac{R_k}{1-\epsilon_{\{1,\dots,k\}}} \leq \sum_{l=1}^{n}\left[Ah(\pi(l)) + (1-\pi(l))L\right] + o(n) \tag{58}$$

The RHS of (58) is separable in terms of $\pi(l)$ and its maximum can be computed via standard derivative arguments. In fact, the maximum in the RHS of (58) is achieved for $\pi(l))=\frac{1}{1+2^{L/A}}$ for $l=1,\dots,n$ which yields

$$n\sum_{k=1}^{N}\frac{R_k}{1-\epsilon_{\{1,\dots,k\}}} \leq nA\log_2(1+2^{L/A}) + o(n) = nL + nA\log_2(1+2^{-L/A}) + o(n) \tag{59}$$

Dividing by n, taking a limit as $n\to\infty$ and using the inequality $\ln(1+x)\leq x$, for any $x>0$, yields

$$\sum_{k=1}^{N}\frac{R_k}{1-\epsilon_{\{1,\dots,k\}}} \leq L + 2^{-L/A}A \tag{60}$$

Repeating the above procedure for an arbitrary permutation σ on $\mathcal{N}$ produces

$$\sum_{k=1}^{N} \frac{R_k}{1-\epsilon_{\{1,\ldots,k\}}} \leq L + 2^{-L/A_\sigma} A_\sigma$$

where $A_\sigma = \sum_{k=1}^{N} \frac{1}{1-\epsilon_{\{\sigma(1),\ldots,\sigma(k)\}}}$ and since the last inequality must be true for all permutations σ, the proof is complete.

C. Proof of Lemma 18

Fano's inequality implies

$$n\sum_{k=1}^{j} R_k = H(W_{[1,j]}) = I(W_{[1,j]}; \boldsymbol{Y}^n_{[1,j]}, \boldsymbol{Z}^n) + o(n) \tag{61}$$

with

$$\begin{aligned} I(W_{[1,j]}; \boldsymbol{Y}^n_{[1,j]}, \boldsymbol{Z}^n) &= \sum_{l=1}^{n} I(W_{[1,j]}; \boldsymbol{Y}_{[1,j]}(l), \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) \\ &= \sum_{l=1}^{n} \Big[I(W_{[1,j]}; \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) + I(W_{[1,j]}; \boldsymbol{Y}_{[1,j]}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}_{[1,j]}(l) \Big] \\ &\overset{it.1}{=} \sum_{l=1}^{n} \Big[I(W_{[1,j]}; \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) + I(W_{[1,j]}; \boldsymbol{Y}_{[1,j]}(l), \boldsymbol{Z}_{[1,j]}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}_{[1,j]}(l) \Big] \end{aligned} \tag{62}$$

Applying the chain rule twice with different order yields

$$\begin{aligned} I(W_{[1,j]}; \boldsymbol{Z}(l), X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) &= I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) + I(W_{[1,j]}; \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, X(l)) \\ &= I(W_{[1,j]}; \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) + I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}(l)) \end{aligned} \tag{63}$$

and since $\boldsymbol{Z}(l)$ is independent of all previous random variables *given* $X(l)$, (63) yields

$$I(W_{[1,j]}; \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) = I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) - I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}(l)) \tag{64}$$

Furthermore, since knowledge of $X(l)$ implies knowledge of $F(l)$, it holds

$$\begin{aligned} I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) &= I(W_{[1,j]}; X(l), F(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) \\ &= I(W_{[1,j]}; F(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) + I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, F(l)) \end{aligned} \tag{65}$$

Combining (64), (65) yields

$$\begin{aligned} I(W_{[1,j]}; \boldsymbol{Z}(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) = I(W_{[1,j]}; F(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}) + I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, F(l)) \\ - I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}(l)) \end{aligned} \tag{66}$$

Defining the set $\mathcal{Z}_{[1,j]} = \big\{ \boldsymbol{Z}_{[1,j]} : \boldsymbol{Z}_{[1,j]} \neq (\varnothing, \ldots, \varnothing) \big\}$, we can compute

$$\begin{aligned} I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}(l)) &= \sum_{\boldsymbol{z} \in \mathcal{Z}_{[1,j]}} I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \boldsymbol{Z}(l) = \boldsymbol{z}) \Pr(\boldsymbol{Z}(l) = \boldsymbol{z}) \\ &= \sum_{\boldsymbol{z} \in \mathcal{Z}_{[1,j]}} I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, \hat{\boldsymbol{Z}}(l) = \boldsymbol{z}, F(l) = 1) \Pr(\hat{\boldsymbol{Z}}(l) = \boldsymbol{z}) \Pr(F(l) = 1) \\ &= \sum_{\boldsymbol{z} \in \mathcal{Z}_{[1,j]}} I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, F(l) = 1) \Pr(\hat{\boldsymbol{Z}}(l) = \boldsymbol{z}) \Pr(F(l) = 1) \\ &= I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, F(l) = 1) \Pr(F(l) = 1) \\ &= I(W_{[1,j]}; X(l) | \boldsymbol{Y}^{l-1}_{[1,j]}, \boldsymbol{Z}^{l-1}, F(l)) \end{aligned} \tag{67}$$

where we exploited the independence of $\hat{\boldsymbol{Z}}(l)$ from all variables up to slot l and used the facts that $F(l)=0$ implies $X(l)=\varnothing$ and $\sum_{\boldsymbol{z}\in\mathcal{Z}_{[1,j]}}\Pr(\hat{\boldsymbol{Z}}(l)=\boldsymbol{z})=1$.

To manipulate the last term in (62), we define the set $\tilde{\mathcal{Z}}_{[1,j]}=\big\{\boldsymbol{Z}_{[1,j]}:\boldsymbol{Z}_{[1,j]}\neq(\varnothing,\dots,\varnothing),(*,\dots,*)\big\}$. In words, $\tilde{\mathcal{Z}}_{[1,j]}$ is the set of feedback vectors in which at least one user in $\{1,\dots,j\}$ successfully receives the transmitted symbol and sends back ACK. Notice that, for any $\boldsymbol{z}\notin\tilde{\mathcal{Z}}_{[1,j]}$, the event $\{\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z}\}$ implies full knowledge of $\boldsymbol{Y}_{[1,j]}(l)$. It now holds

$$\begin{aligned}
&I(W_{[1,j]};\boldsymbol{Y}_{[1,j]}(l),\boldsymbol{Z}_{[1,j]}(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},\boldsymbol{Z}_{[1,j]}(l))\\
&=\sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j]}} I(W_{[1,j]};\boldsymbol{Y}_{[1,j]}(l),\boldsymbol{Z}_{[1,j]}(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z})\Pr(\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z})\\
&=\sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j]}}\Big[H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z})-H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},\boldsymbol{Y}_{[1,j]}(l),\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z})\Big]\Pr(\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z})\\
&=\sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j]}}\Big[H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},F(l)=1,\hat{\boldsymbol{Z}}_{[1,j]}(l)=\boldsymbol{z})-H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},\boldsymbol{Y}_{[1,j]}(l),F(l)=1,\hat{\boldsymbol{Z}}_{[1,j]}(l)=\boldsymbol{z})\Big]\\
&\times\Pr(\hat{\boldsymbol{Z}}_{[1,j]}(l)=\boldsymbol{z})\Pr(F(l)=1)\\
&=\sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j]}}\Big[H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},F(l)=1)-H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},F(l)=1,X(l))\Big]\Pr(\hat{\boldsymbol{Z}}_{[1,j]}(l)=\boldsymbol{z})\Pr(F(l)=1)\\
&=(1-\epsilon_{\{1,\dots,j\}})(1-\pi(l))I(W_{[1,j]};X(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},F(l)=1)
\end{aligned}\tag{68}$$

In the transition from the third to the fourth line of (68), we used the event identity $\{\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z}\}=\{\hat{\boldsymbol{Z}}_{[1,j]}(l)=\boldsymbol{z},F(l)=1\}$, which is valid for any $\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j]}$, while in the transition from the fourth to the fifth line we used the facts that $\hat{\boldsymbol{Z}}_{[1,j]}$ is independent of all variables up to l (including $F(l)$, $X(l)$) and knowledge of $\hat{\boldsymbol{Y}}_{[1,j]}(l)$, $\boldsymbol{Z}_{[1,j]}(l)=\boldsymbol{z}$ implies knowledge of $X(l)$ for any $\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j]}$.

Inserting (68), (67), (66) into (61), via (62), and using item 5 in Lemma 17 produces

$$\begin{aligned}
n\sum_{k=1}^{k}R_k&\leq o(n)+\sum_{l=1}^{n}\Big[I(W_{[1,j]};F(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1})+(1-\epsilon_{\{1,\dots,j\}})(1-\pi(l))I(W_{[1,j]};X(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},F(l)=1)\Big]\\
&\leq o(n)+\sum_{l=1}^{n}\Big[h(\pi(l))+(1-\epsilon_{\{1,\dots,j\}})(1-\pi(l))I(W_{[1,j]};X(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1},F(l)=1)\Big]
\end{aligned}\tag{69}$$

where we used the inequality $I(W_{[1,j]};F(l)|\boldsymbol{Y}^{l-1}_{[1,j]},\boldsymbol{Z}^{l-1})\leq H(F(l))=h(\pi(l))$.

D. Proof of Lemma 19

Performing similar manipulations as in the proof of Lemma 18, we can write

$$
\begin{aligned}
n\sum_{k=1}^{j} R_k &= H(W_{[1,j]}) \geq I(W_{[1,j]};\boldsymbol{Y}^n_{[1,j+1]},\boldsymbol{Z}^n) = \sum_{l=1}^{n} I(W_{[1,j]};\boldsymbol{Y}_{[1,j+1]}(l),\boldsymbol{Z}(l)|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1}) \\
&\geq \sum_{l=1}^{n} I(W_{[1,j]};\boldsymbol{Y}_{[1,j+1]}(l)|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},\boldsymbol{Z}(l)) \\
&= \sum_{l=1}^{n} \sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j+1]}} I(W_{[1,j]};\boldsymbol{Y}_{[1,j+1]}(l)|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},\boldsymbol{Z}(l)=\boldsymbol{z})\Pr(\boldsymbol{Z}(l)=\boldsymbol{z}) \\
&= \sum_{l=1}^{n} \sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j+1]}} I(W_{[1,j]};\boldsymbol{Y}_{[1,j+1]}(l)|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},\hat{\boldsymbol{Z}}(l)=\boldsymbol{z},F(l)=1)\Pr(F(l)=1)\Pr(\hat{\boldsymbol{Z}}(l)=\boldsymbol{z}) \\
&= \sum_{l=1}^{n} \sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j+1]}} \Big[H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},\hat{\boldsymbol{Z}}(l)=\boldsymbol{z},F(l)=1) \\
&\quad - H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},\boldsymbol{Y}_{[1,j+1]}(l),\hat{\boldsymbol{Z}}(l)=\boldsymbol{z},F(l)=1)\Big]\Pr(F(l)=1)\Pr(\hat{\boldsymbol{Z}}(l)=\boldsymbol{z}) \\
&= \sum_{l=1}^{n} \sum_{\boldsymbol{z}\in\tilde{\mathcal{Z}}_{[1,j+1]}} \Big[H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},\hat{\boldsymbol{Z}}(l)=\boldsymbol{z},F(l)=1) \\
&\quad - H(W_{[1,j]}|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},X(l),\hat{\boldsymbol{Z}}(l)=\boldsymbol{z},F(l)=1)\Big]\Pr(F(l)=1)\Pr(\hat{\boldsymbol{Z}}(l)=\boldsymbol{z}) \\
&= (1-\epsilon_{\{1,\ldots,j+1\}})\sum_{l=1}^{n}(1-\pi(l))I(W_{[1,j]};X(l)|\boldsymbol{Y}^{l-1}_{[1,j+1]},\boldsymbol{Z}^{l-1},F(l)=1)
\end{aligned}
\tag{70}
$$

where we used again the independence of $\hat{\boldsymbol{Z}}(l)$ from all other variables.

References

[1] G. Paschos, L. Georgiadis, and L. Tassiulas, "Scheduling with pairwise XORing of packets under statistical overhearing information and feedback," *Queueing Systems*, vol. 72, pp. 361–395, December 2012, special issue for Communication Networks.

[2] M. Gatzianas, L. Georgiadis, and L. Tassiulas, "Multiuser broadcast erasure channel with feedback — capacity and algorithms," in *Proc. 4th Workshop on Network Control and Optimization (NetCoop)*, December 2010, pp. 74–81. [Online]. Available: http://arxiv.org/abs/1009.1254

[3] C.-C. Wang, "Capacity of 1–to-K broadcast packet erasure channels with channel output feedback," in *Proc. 48th Annual Allerton Conference*, October 2010, published in IEEE Trans. Inform. Theory, vol. 58, no. 2, Feb. 2012.

[4] S. Athanasiadou, M. Gatzianas, L. Georgiadis, and L. Tassiulas, "XOR-based coding for the 3-user broadcast erasure channel with feedback," in *Proc. WiOpt, 8th International RAWNET Workshop*, May 2012.

[5] L. Georgiadis and L. Tassiulas, "Broadcast erasure channel with feedback — capacity and algorithms," in *Proc. 5th Workshop on Network Coding Theory and Applications*, June 2009.

[6] Y. Sagduyu and A. Ephremides, "On broadcast stability of queue-based dynamic network coding over erasure channels," *IEEE Trans. Inf. Theory*, vol. 55, no. 12, pp. 5463–5478, December 2009.

[7] D. Traskov, M. Médard, P. Sadeghi, and R. Koetter, "Joint scheduling and instantaneously decodable network coding," in *Proc. IEEE GLOBECOM*, December 2009.

[8] L. Georgiadis, G. Paschos, L. Tassiulas, and L. Libman, "Stability and capacity through evacuation times," in *Proc. Information Theory Workshop (ITW)*, September 2012.

[9] C.-C. Wang and D. Love, "Linear network coding capacity region of 2-receiver MIMO broadcast packet erasure channels with feedback," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, July 2012.

[10] L. Czap, V. Prabhakaran, S. Diggavi, and C. Fragouli, "Broadcasting private messages securely," in *Proc. International Symposium on Information Theory (ISIT)*, July 2012.